ROBOT-BASED HAPTIC PERCEPTION AND TELEPRESENCE FOR THE VISUALLY IMPAIRED

A Thesis
Presented to
The Academic Faculty

by

Chung Hyuk Park

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2012

ROBOT-BASED HAPTIC PERCEPTION AND TELEPRESENCE FOR THE VISUALLY IMPAIRED

Approved by:

Professor Ayanna M. Howard, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Magnus Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Wayne Book
School of Mechanical Engineering
Georgia Institute of Technology

Professor Maysam Ghovanloo
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: June 13, 2012

DEDICATION

For my parents, my wife, and my son,

who have filled my life with love, happiness, and inspiration.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Grant No. 0940146. I would like to express my gratitude to my advisor, Dr. Ayanna Howard, for her guidance and advisement, the National Federation for the Blind and the Center for the Visually Impaired in Atlanta, for participation and support for my experiments, and Rayshun Dorsey of WizKids Science and Technology Center for managing the camp and logistics. I would also like to acknowledge each member of the Human Automation Systems (HumAnS) Lab. for their continued support and sharing of insights regarding the approaches presented here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

With the advancements in medicine and welfare systems, the average life span of modern human beings is expanding, creating a new market for elderly care and assistive technology. Along with the development of assistive devices based on traditional aids such as voice-readers, electronic wheelchairs, and prosthetic limbs, a robotic platform is one of the most suitable platforms for providing multi-purpose assistance in human life. This research focuses on the transference of environmental perception to a human user through the use of interactive multi-modal feedback and an assistive robotic platform. A novel framework for haptic telepresence is presented to solve the problem, and state-of-the-art methodologies from computer vision, haptics, and robotics are utilized.

The objective of this research is to design a framework that achieves the following: 1) This framework integrates visual perception from heterogeneous vision sensors, 2) it enables real-time interactive haptic representation of the real world through a mobile manipulation robotic platform and a haptic interface, and 3) it achieves haptic fusion of multiple sensory modalities from the robotic platform and provides interactive feedback to the human user. Specifically, a set of multi-disciplinary algorithms such as stereo-vision processes, three-dimensional (3D) map-building algorithms, and virtual-proxy based haptic volume representation processes will be integrated into a unified framework to successfully accomplish the goal. The application area of this work is focused on, but not limited to, assisting people with visual impairment with a robotic platform by providing multi-modal feedback of the environment.

CHAPTER I

INTRODUCTION

According to the World Health Organization [63], about 285 million people worldwide are classified as visually impaired, and among them, over 39 million individuals are diagnosed as legally blind. For such individuals, the ability to navigate in real-world environments becomes a challenge, and the ability to perceive objects within the environment involves the use of non-visual cues, which typically requires direct contact with the world. Although many assistive devices [29] have been developed and utilized to aid in daily living, the boundary of living is usually limited to the physical surrounding area for individuals with visual impairments. Teleoperation and robotics technology, which are enabling remote operations and telepresence experience, can bring a new form of assistance that can expand the physical boundaries of living and provide new possibilities with multi-functional capabilities.

Advancements in robotic technology have resulted in various forms of robotic support for human lives such as cleaning, entertainment, education, and even medical care. Robotic technology has also introduced new applications in tele-robotics field such as tele-operation [19, 31, 43, 58, 60] and telepresence [28, 51, 79]. Advancement in the tele-robotics area, however, has one common implicit rule that the person using the technology has knowledge and perception superior to the robot, so the person can have complete control over the technology. In part, this is due to the fact that robotic technology is still continuing to mature and has not adapted to encompass the varying levels of skill associated with the human user. This is especially true when compensating for impaired or disabled sensory functionalities. Nevertheless, if combined with robotic sensing capabilities (such as robotic vision) and multimodal

interaction methodlogies (such as auditory and haptic feedback), a robotic system can be further utilized as a gateway to providing multi-purpose assistance including sensory perception.

Assistive technology, on the other hand, has a long history of adopting new techniques for aiding people with disabilities, and has opened many new opportunities in various aspects of living [21, 30, 42, 44, 85]. Traditional aids, such as wheelchairs, canes, Braille readers, and hearing aids, have basically focused on generating passive and close-range sensory aids. Recent assistive devices, including electronic Braille adaptors and computer-screen readers for the visually impaired (VI) or electronic wheelchairs and prosthetic limbs for the motor impaired, have evolved to transfer more information to the user and have granted wider access to daily living for people with impairments, including education and employment [20, 35, 45, 64, 70]. However, most of these tools are designed to work passively with predefined sources of information, and aid in mobility or manipulation to assist in daily living only in specific conditions. What can be added to this list of assistive devices? Which technology can be further developed to support individuals with visual impairments? Can an intelligent and multi-purpose assistive agent help in this domain?

Given the current status of assistive technology and the advancement in robotics, we speculate that a mobile manipulation robotic platform will be a viable solution to serve as a multi-purpose assistive tool for people with visual impairments. Mobile manipulation is a fast growing area [40, 88] due to advances in robotic systems, increases in computing power, and a rising interest in applications for the home and indoor environments. As an assistive device, a mobile manipulator can aid an impaired person with manipulative tasks using its robotic arm and also provide mobility or remote access using its mobile base. Additionally, a mobile manipulator, being a robotic agent, can monitor the environment and form a non-visual representation—such as haptic or auditory feedback—to transfer the in-situ perception of the remote

environment to the user.

With this new concept of designing an assistive robotic system for individuals with visual impairments, this research aims to address the following three issues. First, the robotic system should transfer the robot's condition, such as moving directions and sensor data, to a human user via non-visual modalities. Furthermore, the robotic system should integrate its heterogeneous sensory data to form a realistic and generalized perception of the environment in which it is located, and transfer it to the human user, i.e. the human user should be able to actively sense the remote environment through the robotic system. Last, but not least, the non-visual interaction pathways should be designed to enable the robotic system to learn advanced skills from the human user, expanding its functionalities based on user demand.

To address these challenging issues, this research is composed as follows. Chapter 1 introduces the overall theme of this research and the prior work in assistive technology, assistive robotics, and haptics. Chapter 2 examines the usefulness and effectiveness of a haptic modality in providing robot status to individuals with visual impairments, and compares this with an auditory modality. Chapter 3 expands the concept of haptic feedback to provide environmental perception by enabling 3D haptic exploration through a mobile robot in a remote environment. Chapter 4 applies the haptic rendering algorithms and finite-state machines (FSM) on a dynamic mobile robotic system with an active depth sensor to enable a real-time interactive telepresence system for the visually impaired. Chapter 5 shows an application of our framework where the haptic interface is used as a training device to teach new skills to the visually impaired user using machine learning algorithms. Chapter 6 concludes with a discussion on future work.

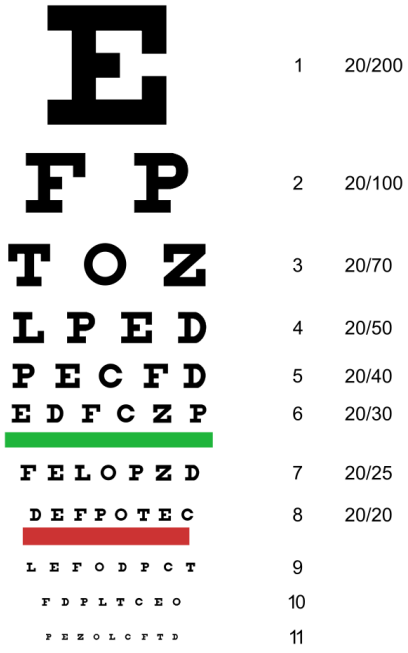


Figure 1: Snellen chart.

1.1 Visual Impairments

Visual impairments (VI) can be categorized into two conditions, low vision and blindness. The causes for visual impairments range from disease, trauma, to congenital or degenerative conditions, and the diagnosis for visual impairment is based on the measurement of visual acuity. The typical level of visual acuity for low vision is 20/60, which means a person with low vision can only recognize an object with size larger than 60mm from 20 feet (6.1 m) away. Figure 1 depicts the Snellen chart that is commonly used to measure visual acuity.

According to Pascolini's study [63] based on the data from WHO (World Health Organization) [59], the global population of those with visual impairments is estimated to be 285 million, among whom 246 million are classified as low vision and 39 million are diagnosed to be blind. In addition, people over 50 years old make up 82% of the VI population. Another report from U.S. Census Bureau (Fact for Features : Americans with Disabilities Act) [12] states that among individuals over

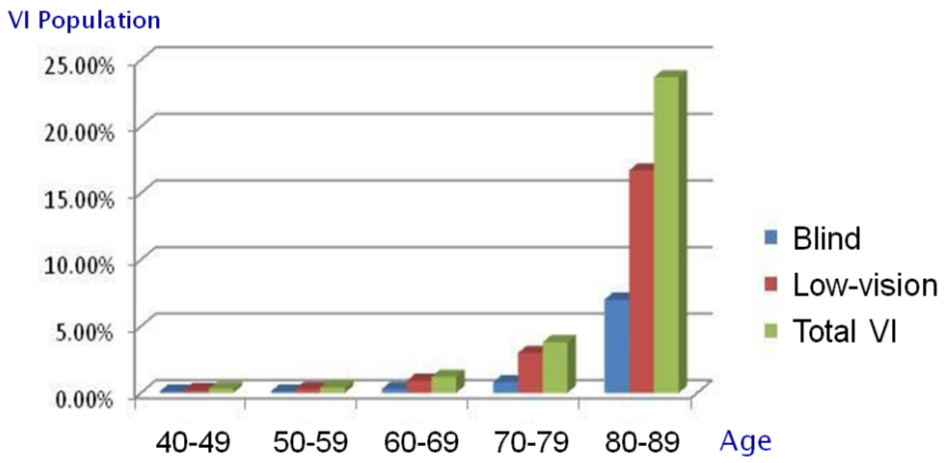


Figure 2: Population of VI among age groups in U.S.A.

Table 1: Population of VI among age groups in U.S.A.

Age Group	*Blindness*		*Low Vision*		*Total VI*	
(Years)	Population	(%)	Population	(%)	Population	(%)
40 - 49	51 K	0.10%	80 K	0.20%	131 K	0.30%
50 - 59	45 K	0.10%	102 K	0.30%	147 K	0.40%
60 - 69	59 K	0.30%	176 K	0.90%	235 K	1.20%
70 - 79	134 K	0.80%	471 K	3.00%	605 K	3.80%
80 -	648 K	7.00%	1,532 K	16.70%	2,180 K	23.70%
Total	937 K	*0.80%	2,361 K	*2.00%	3,298 K	*2.70%

(K = thousand, *: percentage over the whole US Population)

the age of 15, more than 1.8 million people are visually impaired, and the population of individuals with visual impairments per age group increases drastically with age as shown in Figure 2 and Table 1.

1.2 Assistive Technology for the Visually Impaired

For individuals with visual impairments, the ability to navigate in real-world environments becomes a challenge, and the ability to perceive objects within the environment

involves the use of non-visual cues, which typically requires direct contact with the world. The most general assistive navigation device for the visually impaired is the walking cane (Figure 3a). The walking cane is quite inexpensive and easy to use. However, the user has to constantly move the cane around and make physical contact with the environment to perceive the environment in order to navigate. Also, the perceptual feedback is generally limited to large objects and structures that are below waist due to the nature of the cane, and an individual with visual impairment needs to actively use their hands and arms to perceive small objects or structures that are higher than one's waist level.

The traditional method for transferring knowledge and information to individuals with visual impairments, on the other hand, has been through the use of the Braille system (Figure 3b). The Braille system is a well designed tactile language that does not have limitations on the amount of information transferrable. However, it is relatively expensive to generate materials printed with Braille, and it is so complex to learn that a report reveals that fewer than 10 percent of blind people can read it [57].

(a) Walking cane (b) Braille

Figure 3: Traditional aids for VI.

The difficulties and limitations of traditional assistive devices are gradually being resolved through the development of electronic devices and computing technology. For

information transfer and the use of the modern computer and the Internet technology, screen reader softwares (Figure 4a) are substituting the Braille interface. Screen reader software is generally a text-to-speech (TTS) system that can read out any information that the user clicks or sets focus on, and once properly setup the user can easily learn how to use it. Due to this technology, more chances for learning and access to the outside world have become available to the VI community, resulting in increased education and employment opportunities [20, 35, 45, 64, 70].

For navigational aid, electric range sensors such as BAT K-Sonar [1] and Miniguide [2] (as shown in Figure 4) have been developed. These hand-held devices mimic the bat's sonar capability and generate vibratory or sound feedback to notify the user of distance to the object in the direction the user is pointing the device. These sensors are designed to either work with a cane for expanding the spatial range of perception in navigation, or to be used as a standalone device for object finding. The advantages of these devices are that they are low-cost and light-weight, however, the disadvantage lies in the fact that these are passive devices that the user needs to physically activate in order to function.

(a) Screen reader (b) BAT K-Sonar [1] (c) MiniGuide [2]

Figure 4: Examples of modern assistive devices.

Additionally, in the case of sonar-based navigational aids, the user needs to be trained to understand the correlation between feedback patterns and the corresponding spatial information which dynamically changes as the user walks around. Therefore, the added technology cannot work independently, and the user still needs to

vigorously sense different spots and mentally process the data to figure out environmental characteristics.

1.3 Assistive Robots for the Visually Impaired

Efforts to utilize robotic systems for assisting the visually impaired have also been made [42, 85]. Borenstein and Ulrich developed the GuideCane [85] that can sense the environment and guide the individual through the environment with a robotic cane. Utilizing arrays of ultrasound sensors detecting full range of 180° forward, the GuideCane can detect obstacles and rotate its mobile base to adjust its path automatically, safely guiding the user who is holding a cane attached to the robot as shown in Figure 5.

The GuideCane provides autonomous sensing and path guidance for the user so that the user no longer needs to steer around the walking cane to detect obstacles by physically making contacts. The system can be upgraded to expand the sensing range by adding extra sensor arrays facing upward in order to detect objects that cannot be detected by horizontal sensing on the ground. Nevertheless, the user cannot detect or feel small objects that are located above the waist, and feedback for navigation is all it can provide.

Another approach from Kulyukin et al. incorporated a mobile-robot based guidance approach with radio-frequency identification (RFID) devices to assist in navigation and provide richer information than former robotic assistance [42]. By transferring RFID information that is embedded into objects and environmental structures, the user can get both navigational guidance and more detailed object information. Using this concept, Kulyukin et al. developed a RoboCart that can assist the visually impaired in navigating and finding items in a grocery store that has RFID tags installed. Figure 6 depicts the RoboCart system and its functional parts, which include a mobile robotic base, a laser range finder (LRF) for mapping and navigation,

Figure 5: GuideCane: a mobile robotic system for navigational aid for VI [85].

a RFID reader, and a laptop for on-board processing. This system addresses both issues of navigation and object identification, suggesting that a robotic system can be useful and provide various types of assistance. However, the limitation lies with the fundamental assumption that all the environmental settings and objects have to be tagged with RFID.

A more challenging effort was made by Hong [30], who developed a framework of a driving system for the visually impaired. Applying knowledge acquired through autonomously driving in the DARPA Grand Challenge, Hong's group developed a vibro-tactile vest and sensor-fusion algorithms that assist the VI user in driving, by haptically transferring information such as appropriate steering angle, obstacles on

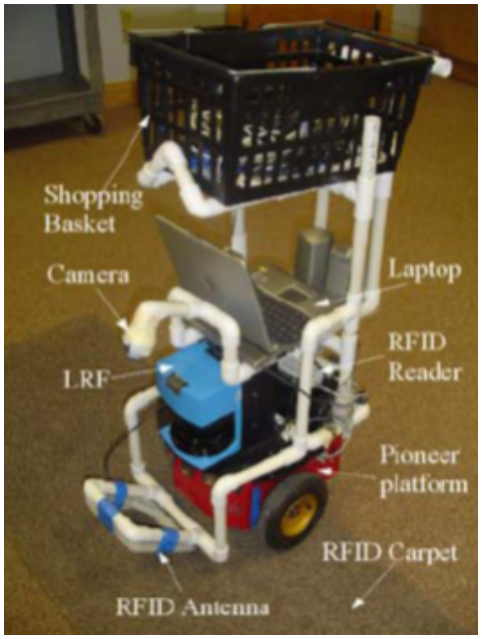


Figure 6: RoboCart [42].

Figure 7: Haptic vest for assisting a blind driver [30].

the road, and speed and acceleration. As shown in Figure 7, the system was able to assist a blind person in driving a car successfully.

These efforts present pioneer studies in the field of assistive robotics in the sense that they intend to enable a person with a visual impairment the ability to maintain independence in specific areas of daily living that were difficult or even impossible before. However, these studies all focus on assistive systems that are directly attached

to the user, limiting the scope of feedback and activity to the vicinity of the user. What is missing here is the assistance with extended range of connectivity for the remote environment, which lead us to the study of telepresence.

As Schoerb summarized in [78], there exist various types and degrees of systems that support telepresence. Among several research concepts, tele-operation and visual feedback systems have been the major areas of research. Michaud et al. developed a robotic telepresence system for home care applications, which is equipped with simple teleoperation capabilities for navigation with visual feedback [50]. Theoretical analysis and modeling for robust teleoperation control has been well studied using the concept of passivity and energy-based models [73]. However, as Figure 8 shows, the actual embodiment of telepresence robotic systems typically rely on visual feedback in providing remote perception to the users.

Therefore, to expand access to the world and enable a visually impaired user to sense and act on a remote environment, this research aims to propose a new solution for remote sensing and telepresence for the visually impaired. The key technology in this solution is haptic modality that can provide non-visual feedback and control methods, and the following section explains the details of haptic technology and its application to this research.

1.4 Haptic Modality for Assisting the Visually Impaired

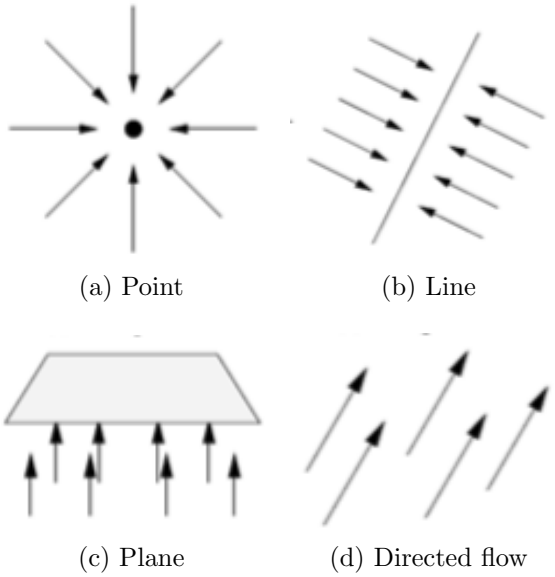
Haptics refers to a field of study dealing with the sense of "touch." Accordingly, "haptic technology" means technology that interfaces with a user through the sense of touch, and "haptic communication" represents the methodology that people or other animals adopt as a means for communication with haptic sensations. In addition, "haptic perception" is defined as the process of recognizing objects through touch. The process for haptic perception is referred as "haptic rendering" or "haptic display."

Figure 8: V-Go telepresence robot.

As shown in Figure 9, haptic forces can be applied to represent four basic primitives: a point, a line, a plane, and a directed flow. Although the haptic force occurs at the exact point of contact between the human body and the haptic interface, the physical configuration of the haptic interface can represent multi-dimensional phenomena of the world—such as static volume in 3D, deformable objects, dynamic flows and forces—with combinations of the four haptic primitives.

1.4.1 Haptic Interfaces

Two well-known and low-cost haptic interfaces commercially available are a force-feedback joystick (Figure 10a) and a vibrating game controller such as the Wiimote



(a) Point (b) Line

(c) Plane (d) Directed flow

Figure 9: Haptic force primitives.

(Figure 10b). A joystick is basically a controller that takes 3D tilting-angle inputs from a human's hand. However, combined with a motor-based actuator that generates a point-force primitive at the axis of the joystick's handle within the surrounding region, the force-feedback joystick can generate two-dimensional haptic feedback while taking a human's control inputs. Game controllers such as the Wiimote adopt a motor-based vibration feedback, which is a kind of directed-force primitive in a cylindrical coordinate system. Through the motor's vibration, it can model virtual contacts or crashing events to the user, which helps the user immerse into a virtual reality environment by letting the user feel virtual events that happens with a virtual character associated to the user.

More advanced haptic interfaces include a multi-dimensional force feedback device such as the SensAble^{TM} Phantom. This interface is capable of generating six dimensional force feedback at a high-frequency rate, and enables haptic rendering of 3D shapes, dynamic simulations, and even representation of textures as depicted in Figure 11a. Integrated with virtual reality and wearable computing technology, a

wearable haptic interface has also been developed as shown in Figure 11b [3]. Advanced research topics using haptic interfaces are described in the following section, where we derive our research ideas and novel contributions.

1.4.2 Haptic-Based Research for Assisting the Visually Impaired

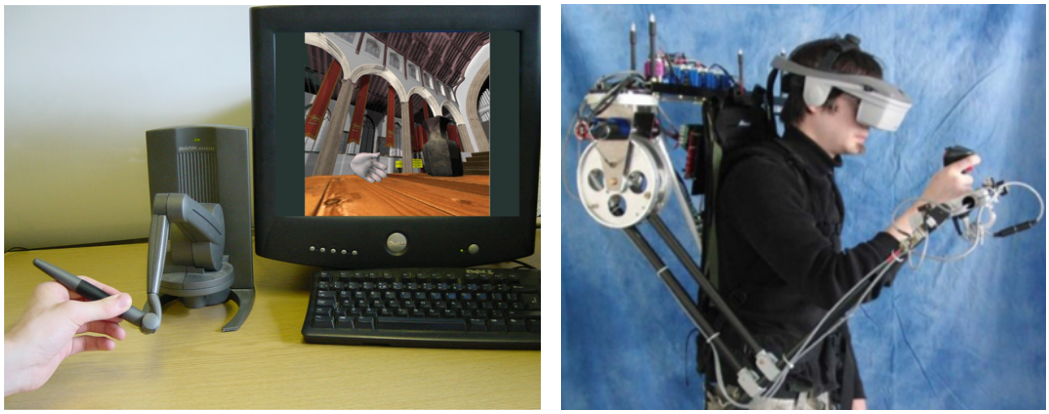
Haptics is a mature field of engineering research, with varying applications in the medical field [6, 44], rehabilitation [15, 44], virtual reality [10], and tele-operation [19, 31, 42, 58, 60]. Its ability to transfer tactile and textural sensations along with force feedback adds another dimension of interaction between the human and the system.

Haptics has been applied in many virtual reality applications [10, 15, 44] showing positive results even for assistive technology [10, 15]. Real-world applications exist such as haptic tele-operated surgery [58] and haptic control of mobile robots [19, 31]. Added to this is the study of efficient haptic visualization and rendering [5, 14, 36, 84] that provide a reliable methodology for presenting haptic data to the user. These related efforts suggest that haptics technology can be a viable solution for an effective mediator that can transfer the details of the environment data as well as handle user's input to the system interactively.

What is lacking here, however, is the haptic display technology based on real-world

(a) Force-feedback joystick (b) Wii remote (Wiimote)

Figure 10: Common haptic interfaces.



(a) SensAble^{TM} Phantom (b) Wearable haptic-interface system [3]

Figure 11: Advanced haptic interfaces.

perception in real-time. Most of the haptic display efforts focus on virtual-reality-based applications, and its merits as a non-visual feedback modality are underestimated for use as an assistive device for the visually impaired. As discussed in [21], the perceptual matching between what is sensed and described by the assistive system and the actual environment can be a major issue, and the mismatch can cause critical limitations as a tool for the visually impaired. Haptic representation of the environment can be the most direct and effective interactive modality for assisting the visually impaired, as derived from the fact that the most common traditional assistive tools for the visually impaired are canes and Braille-notes that stimulate tactile sensations.

In this sense, this research focuses on the problem of utilizing haptic modality to create an interactive channel for communication between a robotic system and the user to aid in controlling and sensing a remote environment. To be more specific, this research aims to provide a novel framework for a visually-impaired user to expand the limitations of one's physical living area through a telepresence robotic assistant using haptic feedback modality. The major contributions to this effort include: 1) study and design of a haptic modality as a non-visual feedback method for in-situ perception transfer, 2) design and evaluation of a vision-based haptic exploration framework that

generates 3D haptic perception for representing real-world 3D spatial information, and 3) design and evaluation of a real-time interactive telepresence system for the visually impaired using an active depth camera, a color vision sensor, audio sensors, a mobile manipulator robotic system, and a haptic interface.

CHAPTER II

HAPTICS FOR NON-VISUAL ENVIRONMENTAL FEEDBACK

The first objective of this research is to evaluate the effectiveness of haptic feedback as a primary means for non-visual feedback in a telepresence robotic assistant for the visually impaired. We design a telerobotic system and a non-visual framework for transferring the robot's in-situ perception. The non-visual feedback signals will enable a person with a visual impairment to access a robot in a remote location and get in-situ information from the robot. The system design includes a human-robot interaction framework that utilizes haptic and auditory feedback to transfer a robot's motion and sensor status to a human user. An experimental design to test the framework is explained, followed by the results and analysis of the experiments.

The remainder of this chapter is composed as follows. Section 2.1 provides an overview of our robotic platform and a multi-modal interaction system, with an elaboration on the different types of feedback and communication flow available. Section 2.2 explains the details of our multi-modal feedback design, and Section 2.3 discuss the application of the multi-modal feedback for the students with visual impairments. Section 2.4 introduces the settings of our experiments and hypotheses of our study. Then, Section 2.5 discusses the results arising from deploying the design during two robotic camps with students with visual impairments. Section 2.6 concludes with an overview of lessons learned with respect to the effectiveness of haptic feedback for telepresence support.

2.1 System Overview

In order to provide a low-cost and intuitive robotic platform, the LEGO Mindstorm^{TM} NXT, as shown in Figure 12, was selected as the hardware platform for this first-stage study. The robots were composed of one LEGO Brick computing block, two motors with wheels and built-in encoders for odometry calculation, two touch sensors to detect user input and bumping incidents, one light sensor to detect a goal on the floor, and one ultrasound sensor to detect an object in front of the robot.

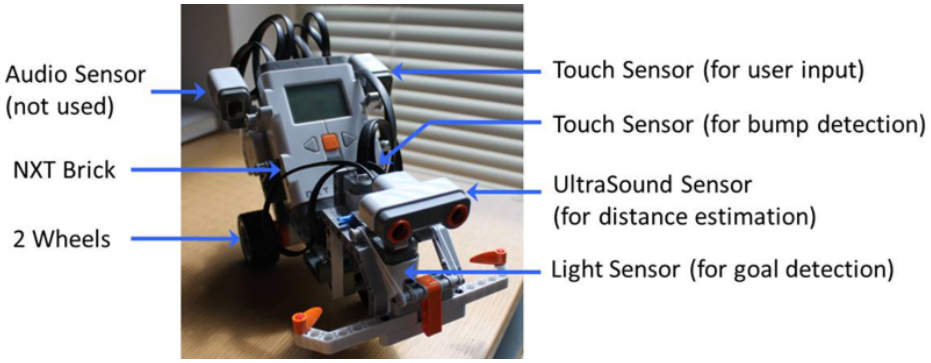


Figure 12: LEGO Mindstorm^{TM} NXT robot.

A Wii remote controller, called Wiimote (Figure 10b), was used as the primary haptic interface between the robot and the user [22]. Wiimote is an interactive game controller that has several buttons for input and a motor for creating vibration feedback. In our system, we use the vibrating function of the motor for haptic feedback generation. The communication between the Wiimote and the LEGO NXT robot is governed by a PC via Bluetooth connection [81], and is discussed in detail in Section 2.1.2.

2.1.1 Feedback Primitives

By providing extra channels for environmental feedback, the user can be aided multimodally in realizing the robot's situation and its movement at the same time. By combining the NXT sensors, our robotic system is able to sense the following environmental elements:

1) Sense of distance traveled by the robot,

2) Sense of direction the robot is turning,

3) Sense of distance to an object located in front of the robot,

4) Sense of whether the robot has reached a goal or not,

5) Sense of whether the robot has bumped into an obstacle.

Using these sensory elements, five feedback primitives were designed to provide in-situ information of the robot via haptic and auditory sensory feedback.

Travel distance feedback: Real-time feedback is provided incrementally based on how far the robot has traveled. Since our robotic platform is small (6in x 4in x 5 in), we generate feedback every 4 inches (≈10cm) while the robot is moving. This provides sufficient associative feeling to the user with respect to sensing of distance.

Turning left/right feedback: Turning is another key movement that plays an important role in robot navigation schemes, yet is difficult to assess for individuals with visual impairments. Different but symmetric signals are designed to correlate the robot's status of turning left or right. The user is also provided with real-time feedback on how many degrees the robot has turned by providing feedback in fixed degree increments In this experiment increments of 45 degrees were selected based on the robot's speed and odometry accuracy.

Object distance feedback: Sensing (and avoiding) an obstacle or a wall is critical in the robot navigation sequence, thus it is important for an individual with a visual impairment to have a corresponding sense of the obstacles that are within close proximity to the robot . An ultrasonic sensor is attached to the front of the NXT robot, and is used to detect obstacles between approximately 5cm and 50cm.

Bump feedback: When the robot bumps into a wall, the mechanical system of the the NXT robot will trigger the lever connected to the bumper to press a touch sensor that signals a bumping event to the remote PC. This information is also transferred to the user to provide feedback on obstacle collisions.

Goal feedback: Once the robot has reached a goal position, a goal event will be transmitted to the user. Based on these feedback primitives and the accumulated knowledge they convey during robot operation, the goal of this work is to provide sufficient feedback to the user to enable them to visualize the sequence of actions executed by the robot with respect to the environment.

2.1.2 Communication and System Architecture

Communication between the user, the robot, and a host computer (used to decode the communication protocols) is accomplished wirelessly via Bluetooth (BT) connectivity. The NXT robot has a BT module that can transmit and receive signals wirelessly at the transfer rate of maximum 11 Kbyte/s. The Wiimote also has a BT module that can be interfaced wirelessly with the host PC. The PC that controls the communication thus opens two BT connections, one with the NXT robot and the other with the Wiimote, handles the signals and manages the multi-modal feedback process. The detailed system architecture is shown in Figure 13. Every time the NXT robot receives sensor requests from the PC, it transmits basic sensor data (such as touch sensor, light sensor, and encoder values) along with the ultrasonic sensor value that specifically uses an I^2C interface. Once the PC receives the sensor values, sensor noise is filtered out using case-based knowledge on the encoder value patterns. The de-noised sensor data is then incorported into robotic state estimate, such as speed and heading, and is processed in the localization module, which estimates the distance of travel and degrees of turn. After the integration of sensor values and estimation of robot status, the feedback handler module decides when to generate which types of feedback to the user. Finally, haptic feedback is generated on the Wiimote that is connected via BT, while auditory feedback is generated on a PC speaker system.

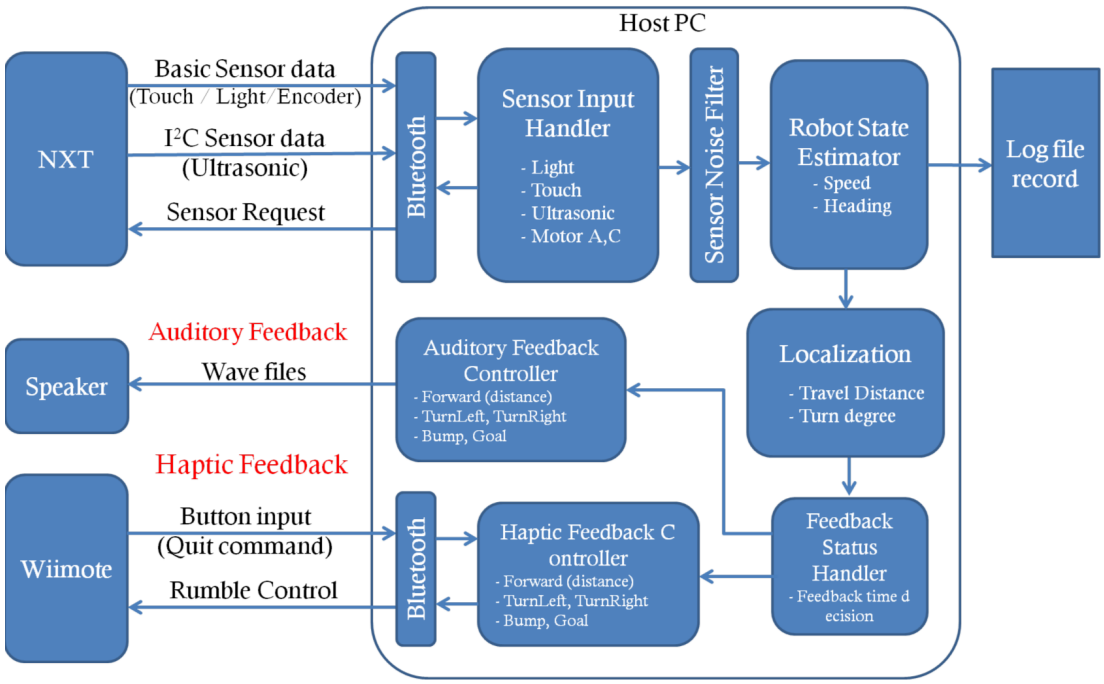


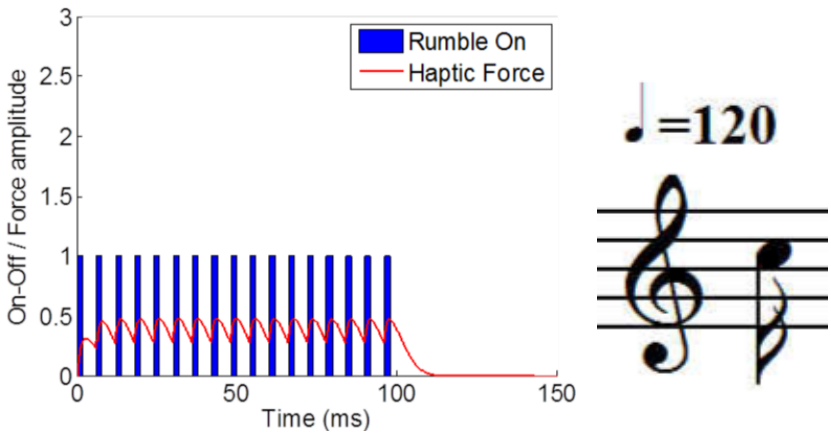
Figure 13: System Architecture and Communication Flow

2.2 Multi-Modal Feedback Design

2.2.1 Feedback Description

In this section, we describe the method in which the five environmental feedback primitives are implemented. All feedback primitives are designed for both haptic and auditory feedback, and are transferred to the user in real-time while the robot is moving. Time durations for the haptic and auditory feedback signals are specified below. Commonly the haptic feedback takes a longer time since haptic signals are perceived in low-frequency, and requires more idle time for multiple signal recognition.

In order to create multi-level sensitivity for haptic feedback, we incorporate a pulse-width modulation (PWM) technique [11], which enables us to control the force of vibration by changing the on/off duty cycle of the vibrating motor. Since the motor will continue to vibrate (with decaying order) even after the motor is cycled off, we used skewed Gaussian graphs to estimate the force envelope required for a single shot motor on/off. This allowed us to design an estimate of the vibration model despite



(a) (b)

Figure 14: Feedback signals for NXT robot traveling forward: (a) Haptic feedback and (b) auditory feedback.

the fact that the Wiimote's actual motor specifications and software controller are not readily available. Using this approach, we can estimate the energy transfer to the Wiimote (needed to enable motor vibration). For a motor-control pulse wave, $f(t)$, with a high value $y_{max} = 1$, a low value $y_{min} = 0$, and a duty cycle D $(0 < D < 1)$ over time period T, the estimate of the output haptic force y is as described in Eq.1.

$$f(t) = \begin{cases} y_{max} = 1\,(on),\ for\ \ 0 < t < D \cdot T \\ y_{max} = 0\,(off),\ for\ D \cdot T < t < T \end{cases} \tag{1}$$

Thus, by controlling the duty cycle on/off, we can change the strength of the feedback force provided by the Wiimote.

Travel Distance feedback: The feedback signals associated with distance of travel is composed of a basic haptic feedback element of a weak vibration for 100ms, and a sound of playing a quarter note of "do" (in octave C4) on the piano lasting 28ms for auditory feedback. 200ms of idle time is added to distinguish between signals in case multiple consecutive signals are transferred. The feedback graphs for both haptic and auditory signals are illustrated in Figure 14.

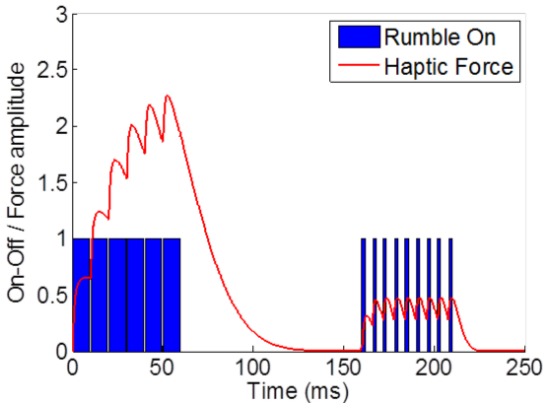
Turn Left/Right Feedback: For representing the left-turn signal, we use a

combination of a high-frequency feedback element (60ms) followed by an ideal time of 100ms and a low-frequency feedback element (50ms) to form a left-skewed haptic signal profile. For auditory signals, a quarter note of "sol" followed by a quarter note of "mi" was played and recorded from the piano, lasting 37ms. For representing the right-turn signal, we use a combination of a low-frequency feedback element (60ms) followed by an ideal time of 100ms and a high-frequency feedback element (50ms), to form a right-skewed signal haptic profile. For auditory signals, a quarter note of "mi" followed by a quarter note of "sol" was played and recorded, lasting 37ms. The feedback graphs for both haptic and auditory signals are illustrated in Figure 15.

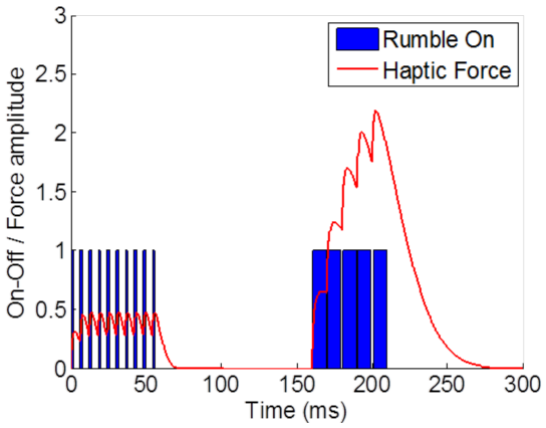
Object Distance feedback: For object distance feedback, signals are designed to alert the user of increasing danger, which could result in crashing into a wall or an object. A high-frequency feedback element is generated for 100ms for haptic feedback, and the sound of playing a quarter note of C-chord harmony (in octave C4 and C5) on the piano is recorded for 37ms. The resolution of the estimated distance to the object is 1 cm over a 5cm $\sim$ 50 cm range. Considering the speed and the size of our NXT robot, feedback was generated every 5cm when the robot is within 15cm of an object. The graphs for haptic and auditory feedback for object distance are depicted in Figure 16.

Bump feedback: When the robot bumps into a wall, a unique and immediate feedback (e.g. a music discord) will be relayed to the user. The haptic feedback is composed of a strong high-frequency feedback element for 500ms, and the auditory feedback was generated from a glass-breaking cacophony for 87ms to address the negative-context of the event. This feedback is intended to provide a strong negative feedback to the user, as shown in Figure 17.

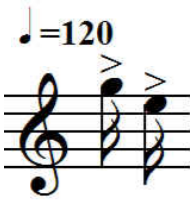
Goal feedback: When the robot reaches a goal identified by a sharply contrasted image overlaid on the floor, the light sensor at the bottom of the robot detects the event. As a positive-context feedback, the corresponding feedback includes a joyful



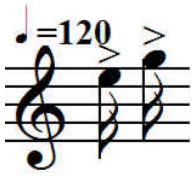
(a)



(b)

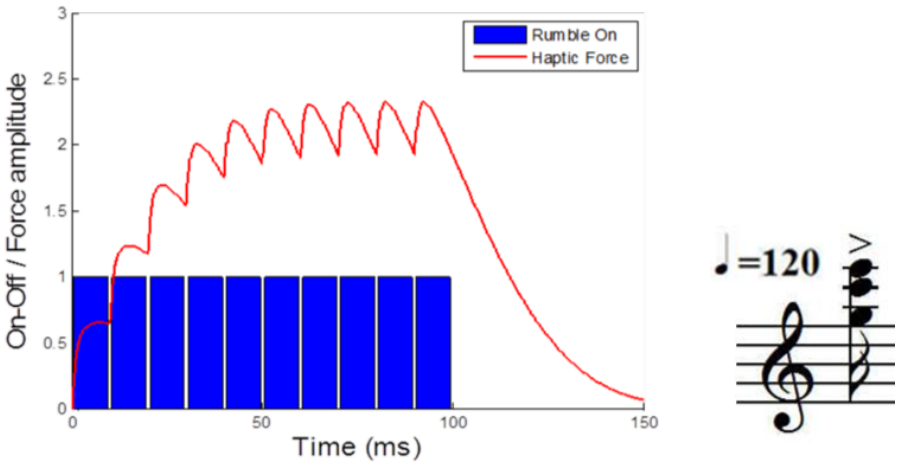


(c)



(d)

Figure 15: Feedback signals for NXT robot turning left/right: (a,b) Haptic feedback and (c,d) auditory feedback.



(a) (b)

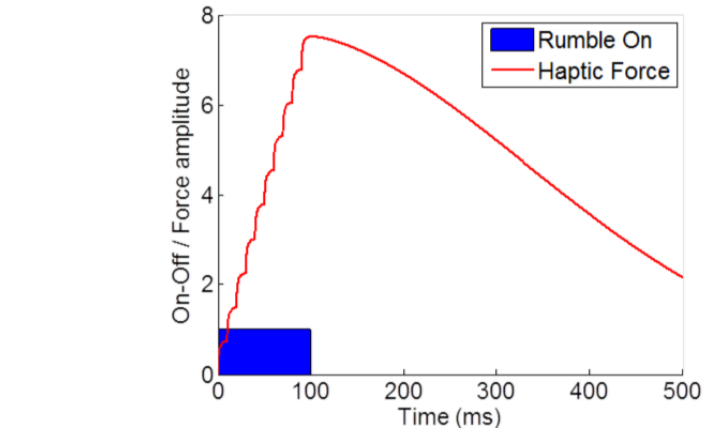
Figure 16: Feedback signals for NXT robot sensing an obstacle in front: (a) Haptic feedback and (b) auditory feedback.

sound (a quarter note of C-chord melody in octave C4 followed by a quarter note of C-chord harmony in octave C4 and C5 for 127ms) and a cheering set of vibrations (fanfare-like haptic vibrations) as illustrated in Figure 18.

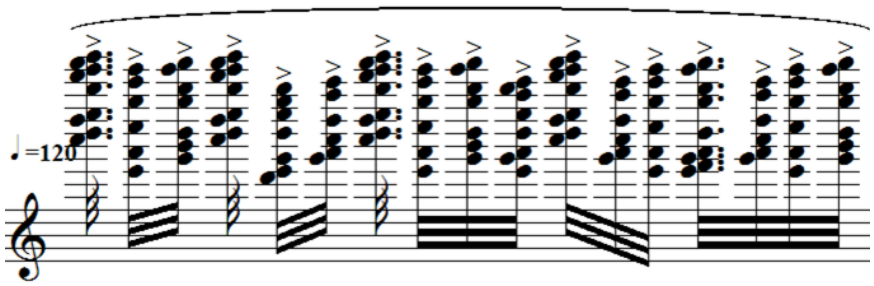
2.3 Application of Multi-modal Feedback for Students with Visual Impairments

Robots can make good companions for individuals with impairments, whether they have visual, auditory, or motor impairments. The robot can assist the user by aiding in sensing capabilities or increase the performance of daily living by jointly working with the person to accomplish common tasks. However, until the day comes that the robot is equipped with perfect autonomous behaviors, the user needs the tools necessary to program or instruct the robot in some intuitive way.

Programming, by itself, can be a daunting task even for the average person. For individuals with impairments, additional challenges arise, such as sensing the environment or recognizing the situation in order to provide a solution to the robot. This issue can be a major hindrance for enabling individuals with impairments in using robotic assistance, further complicating the robot programming task. To overcome



(a)

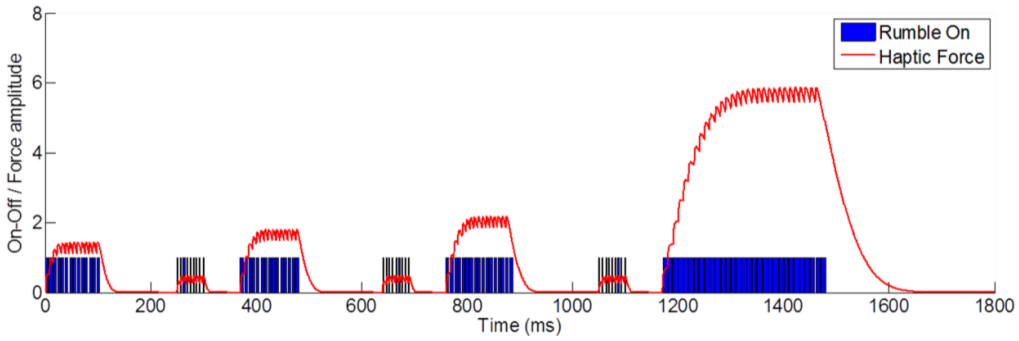


(b)

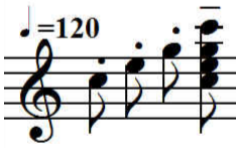
Figure 17: Feedback signals when NXT robot bumped into a wall or an obstacle: (a) Haptic feedback and (b) auditory feedback.

this barrier, non-visual methodologies such as screen reader softwares and haptic / auditory feedback interfaces are being incorporated to grant more access for people with visual impairments to computer and robotic technologies.

Based on this idea, efforts have been made to utilize robotic systems for STEM (Science, Technology, Engineering, and Mathematics) education for students with visual impairments. As a prior work, Ludi organized a robotic programming camp in which students with visual impairments were taught how to program a robot [45]. However, the primary method of sensing the environment was through the participants' direct tactile sensing. i.e. touching, so the students are supposed to follow the robot while physically feeling what the robot does. In contrast. by providing remote telepresece capability, visually-impaired students can program their robot and validate its output without roaming in an unknown environment (Figure 19). With this



(a)



(b)

Figure 18: Feedback signals when NXT robot reached a goal: (a) Haptic feedback and (b) auditory feedback.

framework, the students can sit at their computer console or comfortable chairs and feel the robot moving with haptic and auditory interfaces.

(a)

(b)

Figure 19: Programming and testing a robot in camps for students with visual impairments. (a) Testing with direct tactile sensing, and (b) testing with multi-modal feedback (Wiimotes are marked with circles).

As such, in the next section, we discuss the process for enabling students with visual impairments to validate the program sequence of a robotic system operating in the real world using our multi-modal feedback approach. Using this multi-modal

sensory feedback approach, participants are taught to program and test their own robot to accomplish varying navigation tasks. We discuss and analyze the implementation of the method as deployed during two summer camps in 2010 for middle-school students with visual impairments.

2.4 Experiment Design

In traditional programming processes, we tend to utilize visual feedback to enable writing/compiling of a program, evaluation of the program output, and debugging the program based on this output. Transitioning this to the traditional robot programming processes involves expanding these steps to

1. Writing the program based on the robot command set (library),
2. Compiling the program,
3. Downloading the code onto the robot,
4. Running the code, and
5. Adapting the program based on evaluation of the robot actions.

Again, in traditional settings, these steps tend to be highly visual. Our goal is therefore to utilize additional feedback mechanisms to facilitate programming of robots for students with visual impairments. Our strategy involves partitioning the interaction space into three primary feedback components: 1) interaction/feedback during programming, 2) interaction/feedback during program execution, and 3) interaction/feedback after program execution. Based on these components, we begin by postulating three hypotheses:

- Hypothesis 1: Existing computer accessibility technology (e.g., computer screen readers and magnifiers) can be modified and integrated to provide sufficient feedback for students with visual impairments to enable the programming process.
- Hypothesis 2: Correlating haptic and/or audio feedback with real-time program execution can provide sufficient feedback to enable students with visual

impairments to visualize their programmed robot sequences.

- Hypothesis 3: Enabling automated verbal feedback to summarize program output after completion can provide sufficient feedback to enable students with visual impairments to understand changes that maybe required in their program.

Hypothesis 1 and 3 are designed to study issues other than haptic modality itself, and the detailed analysis is presented in our journal article [34]. In this chapter, we focus on Hypothesis 2 to study the effectiveness of haptic modality.

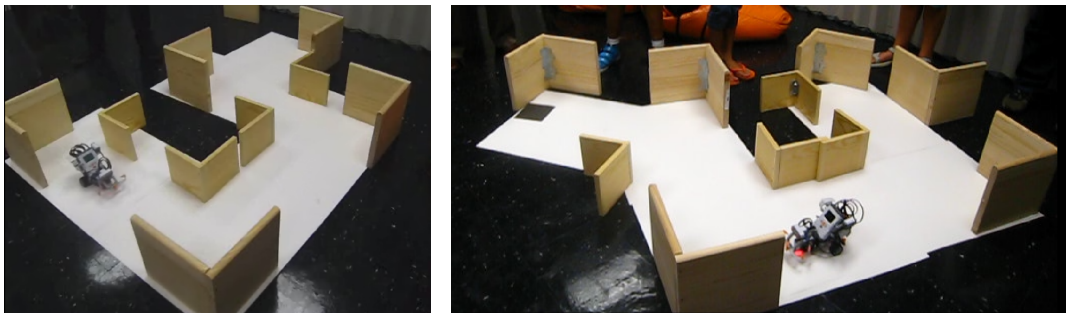
Using the platform described in Chapter 2.1, a teaching protocol was developed that utilized screen readers to relay textual information displayed on the computer console and a lesson plan that provided 1-on-1 instruction on basic programming syntax, compiling and downloading program to the robot, etc. The teaching protocol used was a modification of the NXT resources [46]. The robots for this session were pre-built for the students to provide identical hardware platform for all participants. Given a programmed sequence, we now focus on validating the robot's performance by enabling visualization of the control sequences as they are executed by the robot.

2.5 Results

The aforementioned multi-modal interaction approach to providing enviromental feedback was implemented for two camps for children with visual impairments, during which students were challenged to learn to program a robot. For the first half of the programming session, the students were introduced to the Lego NXT system and basic programming skills, which would enable their robot to perform simple navigation tasks. A robot programming API was provided with commands for moving forward, turning left/right, and through combining commands the students programmed the robot to maneuver through a certain shaped path. During the second half of the session, the students were given tasks to program their robots to navigate through

two different mazes, as depicted in Figure 20. For the maze session, we first provided them with a pre-programmed robot that was capable of succesfully navigating the maze. While the precoded robot with the solution file was navigating though the environment, the students were provided with haptic and/or audio feedback. The students then took notes of what signals they perceived, and based on this knowledge of the environment and the robot's movement, the students programmed their own robots to finish the maze.

Throughout the two camps, a total of 14 students participated in the maze sessions. 10 of them solved both the first maze and the second maze, and 4 of them only had time to solve the first maze due to different schedules. For the maze session, participants were grouped into teams of one or two members. The two-member teams were given only one feedback type per person, that is, one member received haptic feedback while the other received auditory feedback. Therefore, a total of two preprogrammed runs were performed for the two-member teams. During each run, the student feeling the robot's movement (with the other team member absent) would report what they felt their robot was doing (i.e. moving forward 10cm, turning 45 degrees, etc.) and an assisting person would take notes for them. For the one-member teams, the single member was provided with both haptic and audio feedback simultaneously. As in the other team scenario, the students would vocalize what they estimated their robot to be doing, and a person with assistive role would take notes. To form a comparison group, three subjects without visual impairments participated in the experiments.



(a) Maze one (b) Maze two

Figure 20: Two mazes used throughout the camp to test the students' programming skills and the multi-modal feedback system.

Table 2: Post-session Survey Results for Questions 1 and 2

	Survey Questions	Mean Resp.	STD Dev
Q1	How helpful was the Wiimote vibration for understanding the robot's movement?	3.3	1.4
Q2	How helpful was the sound feedback for understanding the robot's movement?	3.3	1.2

2.5.1 User Responses

Questions 1 and 2 of the post-session survey were used to evaluate Hypothesis 2, whether correlating haptic/audio feedback with real-time program execution was sufficient to enable the students to understand their programmed robot sequences. Options for questions 1 and 2 represent a continuous scale with respect to the overall experience and ranged from "very helpful" = 4, "helpful" = 3, "a little helpful" = 2, and "not helpful at all" = 1. Table 2 provides the associated mean response and standard deviation values for each question. Based on the response from the two questions, the students agreed that both the haptic device and audio feedback were helpful in understanding the movement of the robot.

2.5.2 Multi-Modal Perception Results

Seven sets of data were collected from the teams that participated in the first maze, and four sets of data were gathered from the second maze. This data was analyzed and compared with the correct sets of feedback (reference sets) generated from the solution robot. Since some of the subjects missed out or incorrectly perceived a few feedback signals, the recorded results had various lengths, making it impossible to compare the sequences directly. Therefore, we incorporated the dynamic time warping (DTW) algorithm [74] to compare the student-generated sequence with the reference sequence. The DTW is an algorithm for measuring similarity between two sequences of signals (target sequence and reference sequence) which may have been altered in time or speed. The DTW algorithm compares the two sequences progressively to measure the best match, while it inserts or removes signal elements from the target sequence. This algorithm is well applied in audio or video signal matching, and the well-known example of real-world usage is in "speech recognition" for cell-phones.

The DTW comparison results show that the distance (error term) of the subjects' perception ranges from 0 to positive integers, where a 0 represents a perfect match and a positive number reflects the number of mismatches that cannot be warped by addition or subtraction of a sequence. Two examples of DTW matching results are plotted in Figure 21 and Figure 22, and the total data is visualized in Figure 23 and Figure 24 using a Boxplot [49] to show the statistical distribution. We do not have enough statistical data to draw conclusions due to the nature of the sessions with individuals with visual impairments. However, Figure 23 shows that the multi-modal framework using both haptic and audio feedback enables more accurate perception of the user for remote robotic status than the cases of using only one modality, which leads to more effective learning of the students who can acknowledge errors in their programming more clearly.

In order to compare the learning activities of the students with visual impairments to the performance of people without visual impairments, we invited a group of volunteers within our graduate school, instructed each volunteer to face the wall, and recorded their perception from the haptic and auditory feedback. As illustrated in Figure 23, we can see that the mean DTW results (error in perception) from the haptic feedback of people with visual impairments and people without visual impairments are comparable: the deviation of DTW results from auditory feedback overlap by approximately 75% (range $\approx [1, 4]$) between the two groups, and the total distributions also overlap more than 80% as can be seen from the last two boxes of Figure 23 and Figure 24. In both groups, the perception error decreased significantly when both modalities were utilized.

One extra lesson we learned based on observation is that the largest contributing factor to the error was mis-classification between left and right turns. We believe this was caused by the symmetric nature of these generated signals and will be investigated in our future efforts.

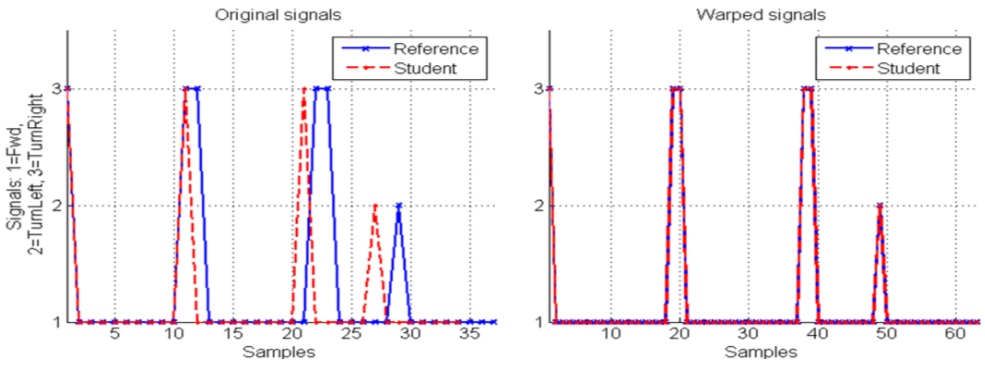


Figure 21: DTW result of Group 1 in Maze #2 with both haptic and auditory feedback

2.5.3 Learning Performance Results

After recording the students' perception of the robot control sequence necessary to navigate through the maze, the participants programmed their robots (one robot per each group) and were challenged to run through the maze until their robot was able

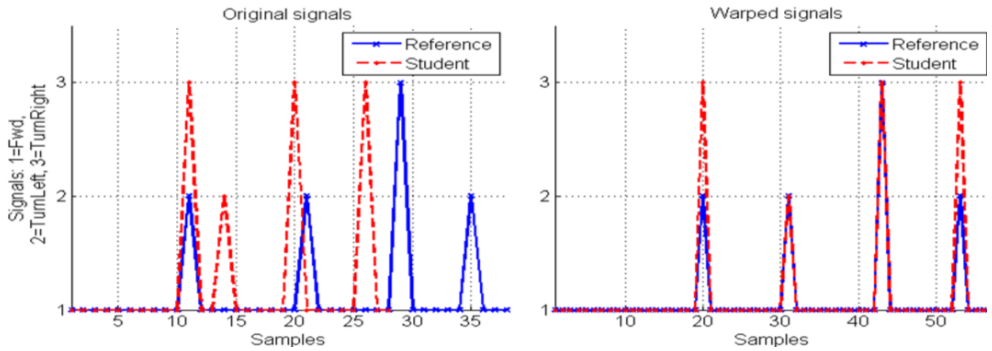


Figure 22: DTW result of Group 2 in Maze #1 with only haptic feedback

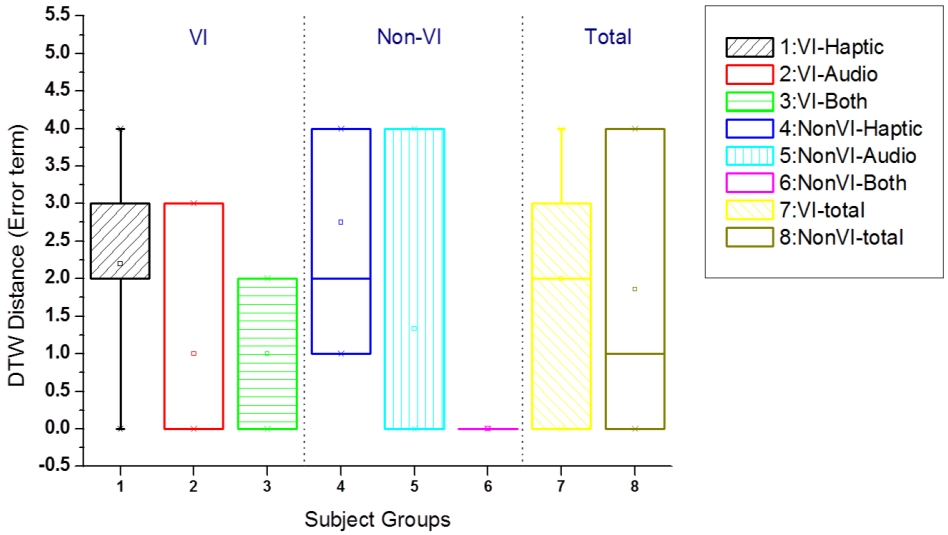


Figure 23: Boxplot of DTW for Maze #1

to successfully navigate through it. The movements of the robots were recorded and analyzed later using the DTW algorithm to assess the learning curves of the group.

Figure 25 shows that for the first maze, (and similar results from the second maze), three teams were able to finish the task in two trials, two teams made it in three trials, one team finished in four trials, and one team finished in eight trials. Comparing with the results depicted in Figure 26 from people without visual impairments, no significantly different results could be found. Thus, to analyze the learning curve more accurately, we calculated the step-wise learning ratio—the ratio of decrease in DTW distances per trial to the initial DTW distance—of every trials. As shown in Figure 27, for each of the mazes, the step-wise learning ratios between subjects

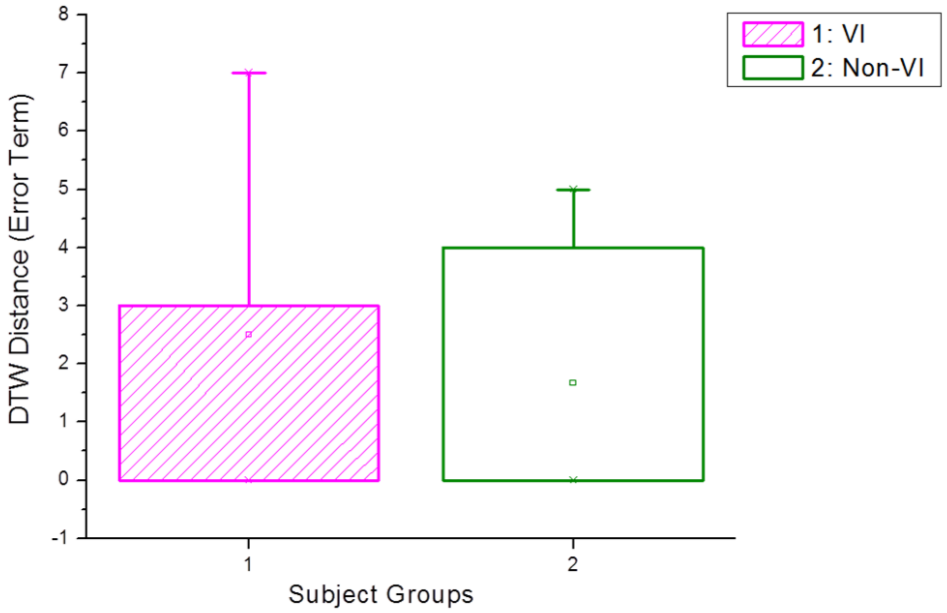


Figure 24: Boxplot of DTW for Maze #2

(teams) with visual impairments and without visual impairments do not show statistically significant differences, which suggest that our multi-modal feedback system is serving its design objectives.

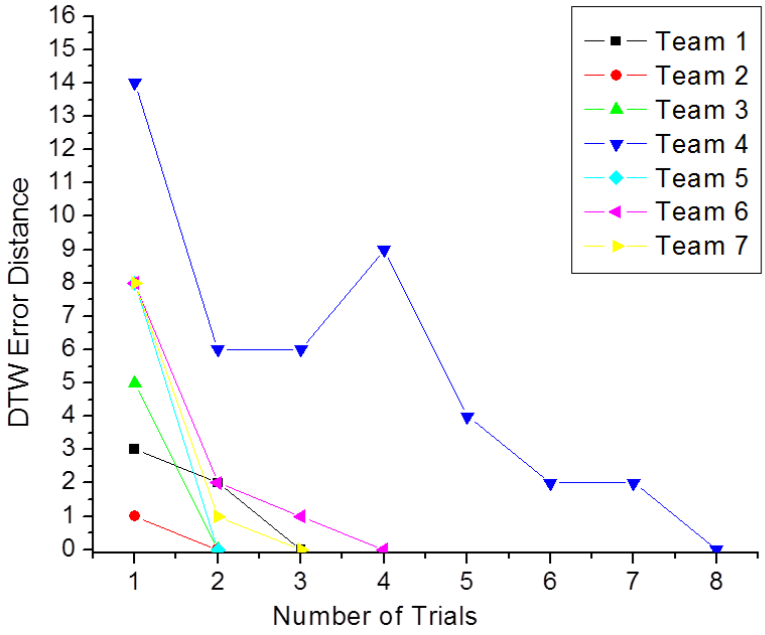


Figure 25: Learning curve for Maze #1.

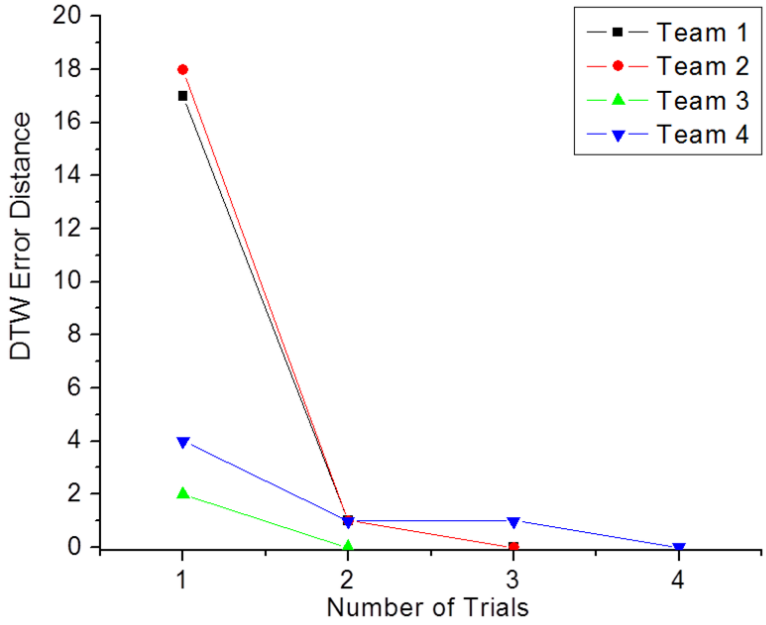


Figure 26: Learning curve for Maze #2.

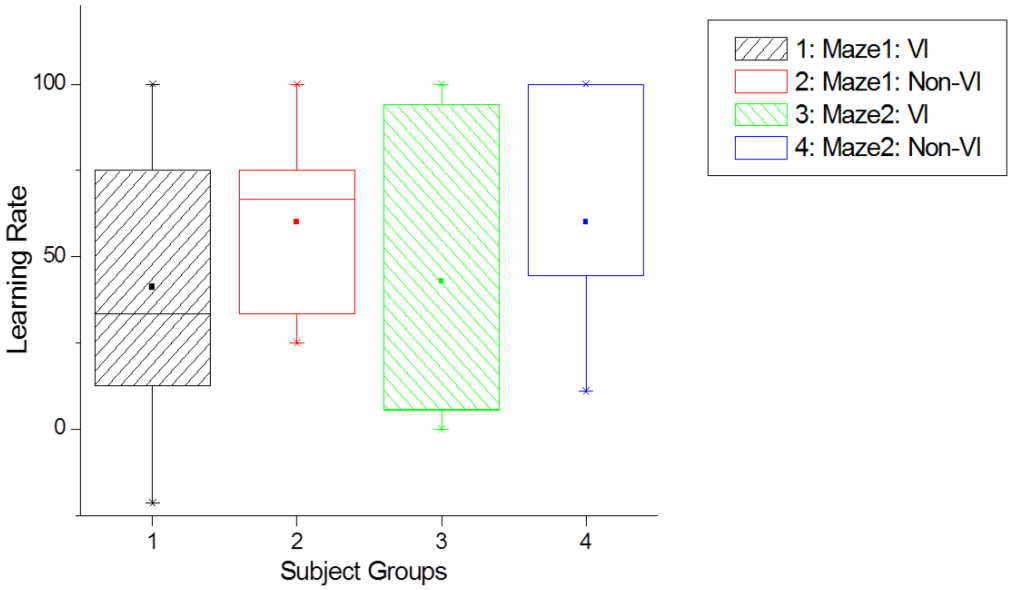


Figure 27: Learning rate.

2.6 Conclusion

In this work, we have proposed a multi-modal framework for transferring environmental feedback of a robot to individuals with a visual impairment. We designed

the hardware system to incorporate mobility and remoteness, with the haptic and auditory feedback signals tailored to the capability of the system. The signals are not additive, but rather consist of a sequence of real-time signals from which the user can build up knowledge of the environment. Results show that people with disabilities can perform even highly visual tasks, such as programming a robot, if provided with extra sensory feedback about the problem and the environment. In the following chapters, we focus on expanding the robotic perception into vision-based sensors—such as a stereo vision sensor, a monocular vision sensor, and an active depth cameras such as the Kinect—and integrating the perception from the sensors into real-time 3D haptic representation of the environment.

CHAPTER III

HAPTIC EXPLORATION FROM ROBOTIC VISION

Given the effectiveness of haptic feedback, as shown in the previous chapter, we now focus on the algorithms needed to enable 3D haptic exploration directly from robotic visual perception. In order to achieve this objective, a haptic linkage is created between an assistive robotic system and a human user by delivering haptic perceptions of a remote environment. More specifically, this research tackles the problem of providing individuals with visual impairments with a sense of the world based on the environmental data acquired by the perceptual system of a mobile robotic platform. To accomplish this, the user is provided with the ability to haptically explore the real-world environment at multiple scales through the "eyes" of a robotic system.

The challenges that must be addressed by this system involve 1) recognizing the environment in multi-scale dimensions since real-world objects are of various sizes and located at different three-dimensional world positions, 2) developing a system architecture for fast processing of the vision sensor data, and 3) enabling three-dimensional haptic interaction for smooth tactile representation of objects in the environment.

The details of this approach are described in the following sections. Section 3.1 describes the robotic platform used for achieving these research objectives. Section 3.2 explains the vision processing algorithms, with a specific focus on stereo matching and monocular-vision-based stereo processes. Section 3.3 deals with the transformation between 2D perception and 3D map generation, while Section 3.4 details the haptic rendering algorithm proposed in this work that works with point-cloud data to form haptic feedback forces. Finally, Section 3.5 shows the results of each algorithm blocks and discusses the performance.

3.1 System Overview

The role of a mobile manipulation robotic assistant for the visually impaired is to 1) use its sensory devices to perceive the environment and generate a 3D perception model of the environment, 2) transfer the environmental perception to the human user in a non-visual way, and 3) transform human controls on the haptic interface to enable teleoperation and telepresence for the human user. Therefore, the haptic interface in this framework functions in two ways: as a controller for the mobile manipulation system and as a generator for the environmental feedback to the human user. Likewise, the mobile manipulation robotic system becomes a mediator between the user and environment working as both the agent that executes the human operator's commands and the agent that collects the environmental data for haptic exploration of the 3D environment. The basic architecture of this haptic-exploratory mobile manipulation (HEMM) system, as illustrated in Figure 28, consists of a mobile manipulator robotic system, a system controller block, and a human interface block. The mobile manipulator robotic system includes sensory devices such as a monocular camera and a stereo camera. The system controller controls the mobile base and the manipulator of the robotic system by the command of the human operator through the haptic interface, and also access the sensory devices to transfer sensory data to the human interface module. The details are discussed in the following sections.

3.1.1 Human Interface

The human interface module receives and transmits sensory information to the human operator as well as to the robot controller in the HEMM system. We incorporate both haptic and visual feedback channels for interfacing with the human operator. For the haptic interface, we select a 6 DoF haptic device PHANToM Omni (Figure 29a) for providing haptic feedback in the system. This device is capable of taking six-DoF input and generating three-DoF force feedback on the stylus. For the visual interface,

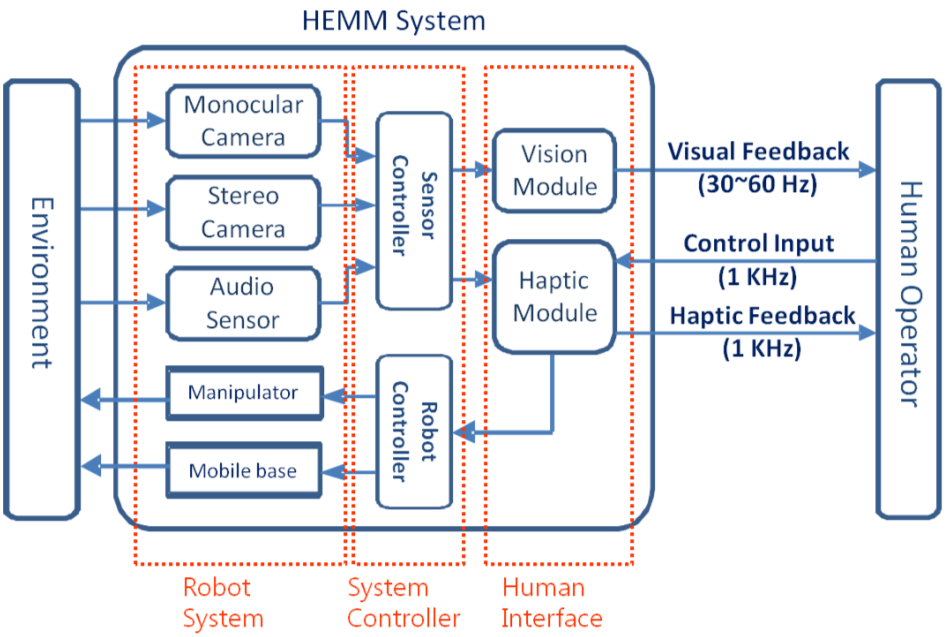


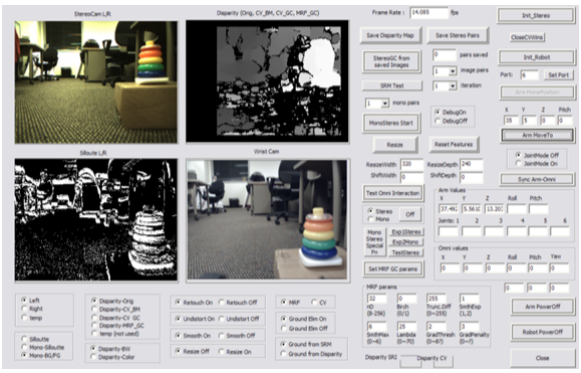
Figure 28: Basic architecture of the HEMM system.

a GUI (graphic-user-interface) has been developed to provide visual feedback from the robotic platfrom, as shown in Figure 29b.

3.1.2 Robotic System

The mobile manipulation robotic system used is the Pioneer3 AT mobile robot with the Pioneer2 Arm. Pioneer 3AT is a 4-wheel drive skid-steer mobile robot system (500 mm x 490 mm x 260 mm (W x D x H)) capable of all-terrain navigation with up to 45deg slope angle, at the speed of up to 0.7 m/sec, and a payload of up to 30 kg. The Pioneer2 Arm is a 5DoF manipulator with a reachable distance of 50 cm from the base joint and a payload of 150g. A stereo camera is positioned at the front of the body frame. A webcam is also mounted on the wrist of the manipulator to provide upclose and dynamic observation of the environment according to the movement of the robotic arm, as shown in Figure 30a.

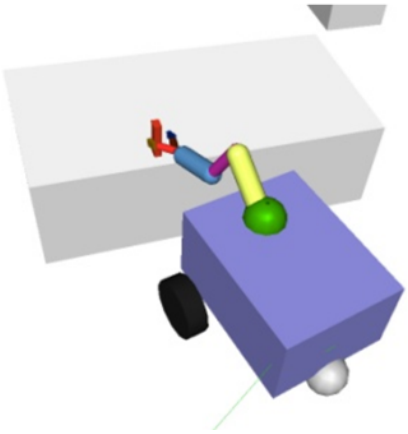
(a) Haptic interface



(b) GUI

Figure 29: Human interface for HEMM system.

(a) HEMM system



(b) Simulated HEMM system

Figure 30: Haptic exploratory mobile manipulation (HEMM) system.

3.1.3 System Controller

For the control of the HEMM system, we utilize several libraries to access the subsystems. The ARIA library [53] is used to develop the algorithms for controlling the mobile manipulator, and the Open Haptics Academic Edition (OHAE) 2.0 [80] and the HAPTIK library [18] are utilized for the control of the haptic interface. Also, the OpenCV (open computer vision) [9] and SVS (SRI vision system) libraries are incorporated for accessing the vision sensors. We also developed a simulated representation of the hardware platform is created in a virtual environment using a rigid multi-body dynamics engine called "srLib" (SNU Robotics Library) [25]. The simulated model consists of a mobile base, a range sensor, and a 4DoF robotic arm, as depicted in Figure 30b.

As for the control algorithms for our HEMM system, a workspace transformation process is needed to map the human input through the haptic device with the actions of the mobile manipulation system. Five workspaces must first be defined: B={set of points in the base frame}, H={set of reachable points in the haptic device's workspace}, E={set of reachable points for haptic exploration in the real/virtual environment with the haptic device}, N={set of control points for navigation of the mobile agent with the haptic device}, and M={set of control points for the manipulation system with the haptic device} as illustrated in Figure 31.

In order to enable a mapping between haptic-based commands and robot operation in the real world, three major workspace transformations exist between the five workspaces (as summarized in Eqs. 2–4). Letting ${}^{B}T_{A}$ represent the transformation matrix for the rotation and translation from A to B, then,

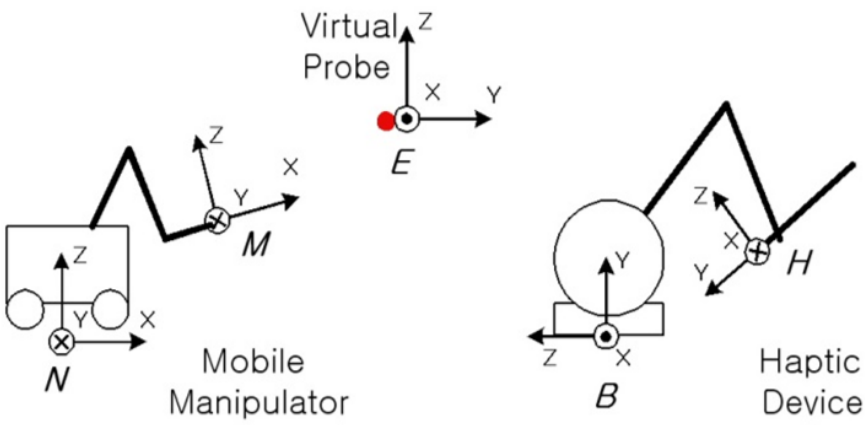


Figure 31: Workspace configurations in a HEMM system.

$$^{E}T_{H} = ^{V}T_{B} \cdot ^{B}T_{H} \tag{2}$$

$$^{N}T_{H} = ^{N}T_{B} \cdot ^{B}T_{H} \tag{3}$$

$$^{M}T_{H} = ^{M}T_{B} \cdot ^{B}T_{H} \tag{4}$$

where $^{V}T_{B}$ is a multi-scale linear mapping between the virtual workspace and the haptic workspace based on the choice of the sensor, $^{B}T_{H}$ is given by the haptic interface, and $^{N}T_{B}$ is a navigation mapping of 3D Cartesian position to a 2D vector of velocity and heading which are translated into the differential speed values for each left/right wheel of the robot.

$^{M}T_{B}$ is a 5 DoF kinematic transformation of the robotic manipulator as shown in Eq.5.

$$^{M}T_{B} = \begin{bmatrix} ^{M}R_{B} & ^{M}l_{B} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{14} \\ \vdots & \ddots & \vdots \\ a_{41} & \cdots & a_{44} \end{bmatrix} \tag{5}$$

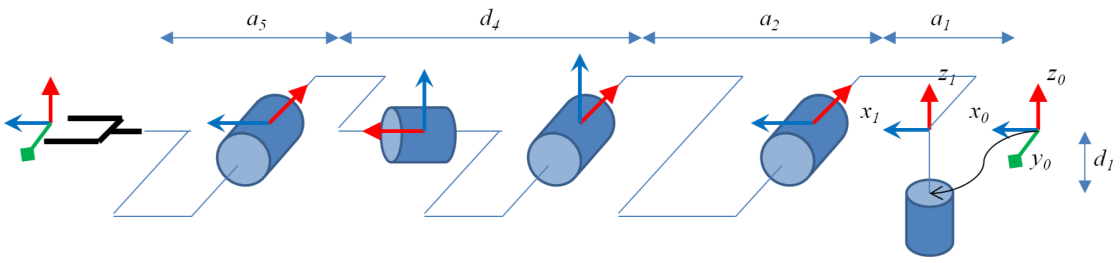


Figure 32: Joint configuration of the Pioneer Arm [41].

Joint	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	d_1(=8.75)	θ_1
2	a_1(=68.75)	90	0	θ_2
3	a_2(=160)	0	0	θ_3+90
4	0	90	d_4(=137.75)	θ_4
5	0	-90	0	θ_5-90
6	a_5(=113.21)	-90	0	0

Figure 33: D-H parameters for the Pioneer Arm.

Here, ${}^{M}R_{B}$ is the rotation matrix , ${}^{M}l_{B}$ the translation matrix, and values of a_{ij} are as defined in Eqs. 6-18 with the joint configurations of the robotic arm defined as in Figures 32 and 33:

$$a_{11} = c_1 s_{23} c_4 s_5 + c_1 c_{23} c_5 + s_1 s_4 s_5 \tag{6}$$

$$a_{12} = -c_1 s_{23} s_4 + s_1 c_4 \tag{7}$$

$$a_{13} = -c_1 s_{23} c_4 c_5 + c_1 c_{23} s_5 - s_1 s_4 c_5 \tag{8}$$

$$a_{14} = A_5(c_1 s_{23} c_4 s_5 + c_1 c_{23} c_5 + s_1 s_4 s_5) + A_4 c_1 c_{23} + A_2 c_1 c_2 + A_1 c_1 \tag{9}$$

$$a_{21} = s_1 s_{23} c_4 s_5 + s_1 c_{23} c_5 - c_1 s_4 s_5 \tag{10}$$

$$a_{22} = -s_1 s_{23} s_4 - c_1 c_4 \tag{11}$$

$$a_{23} = -s_1 s_{23} c_4 c_5 + s_1 c_{23} s_5 + c_1 s_4 c_5 \tag{12}$$

$$a_{24} = A_5(s_1 s_{23} c_4 s_5 + s_1 c_{23} c_5 - c_1 s_4 s_5) + A_4 s_1 c_{23} + A_2 s_1 c_2 + A_1 s_1 \tag{13}$$

$$a_{31} = -c_{23} c_4 s_5 + s_{23} c_5 \tag{14}$$

$$a_{32} = c_{23} s_4 \tag{15}$$

$$a_{33} = c_{23} c_4 c_5 + s_{23} s_5 \tag{16}$$

$$a_{34} = A_5(s_{23} c_5 - c_{23} c_4 s_5) + A_4 s_{23} + A_2 s_2 \tag{17}$$

$$a_{41} = a_{42} = a_{43} = 0, \;\; a_{44} = 1 \tag{18}$$

given that

$$s_i = \sin\theta_i, \;\; c_i = \cos\theta_i \tag{19}$$

$$s_{ij} = \sin(\theta_i + \theta_j), \;\; c_{ij} = \cos(\theta_i + \theta_j) \tag{20}$$

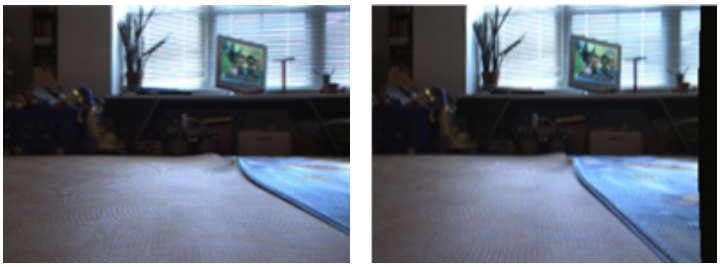
$$A_i = \text{the length of the Arm's joint } i, \;\; i = \{1, 2, 3, 5\} \tag{21}$$

$$\begin{aligned} \theta_i = {} & \text{angle between coordinate frames of joint } (i-1) \text{and joint } i. \\ & \text{(Joint 0 is the arm base and Joint 4 is fixed to limit singularity points)} \end{aligned} \tag{22}$$

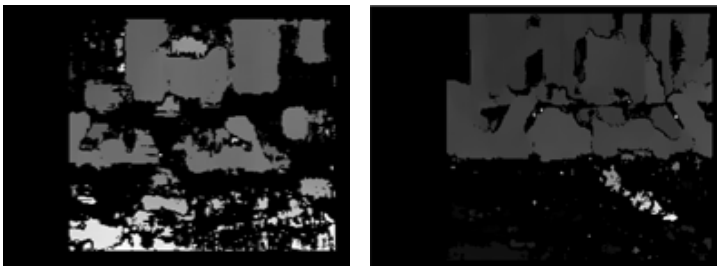
Once we've calculated the transformation sequence, we now need to develop a methodology for dynamically mapping robot sensory data into haptic feedback.

3.2 Stereo Vision Processes

To provide the ability to map sensory data, our HEMM system is equipped with two different camera sensors: a stereo camera in front of the robot and a monocular camera mounted on top of the wrist of the robotic arm. The stereo vision provides large-scale perception of an indoor environment, and the monocular camera enables up-close and detailed viewing of an object. The following sections explain the vision processes and disparity-map generation algorithms for the camera systems used to create a perceptual map of the environment.



(a) Left image (b) Right image



(c) SAD (sum of absolute differences) (d) Block-Matching

(e) Graph-Cut

Figure 34: Several results of stereo disparity-generation algorithms.

3.2.1 Stereo Vision for Large-scale Perception

Stereo-vision processes [37,77,82] have been widely adopted for the spatial recognition of real-world environments. The typical process for stereo-vision algorithms involves creating a depth map (or disparity map) M in which each pixel value is a disparity between corresponding points on two stereo-pair images. The disparity values can be transformed into depth values in a 3D space.

Figure 34 shows the results of a sample indoor environment using three well-known stereo matching processes generating disparity maps. The SAD (sum of absolute differences) [38] is a commonly used algorithm for fast disparity estimation, which calculates intensity differences on matching points along epipolar lines. The Block-Matching algorithm [39] is another widely used algorithm that segments the input images into fixed size blocks and find matching blocks to calculate disparity values. The gray levels in the figures represent the depth differences, with the brighter level meaning the closer distance.

Another algorithm that achieves higher quality results is the graph-cut (GC) energy-optimization method [7, 8] using Markov-random-field (MRF) [83], and based on the evaluation of the different algorithms [77] we choose this GC-MRF method. This method represents a disparity image as a set of MRF variables with two energy functions (representing the cost of estimated disparity and the cost of smoothness with neighboring pixels) and calculates the optimal solution by minimizing the total-energy function.

The energy model for MRF-based stereo-matching algorithm consists of a data-term, E_d, and a smoothness-term, E_s, both represented in the form of energy functions. For a disparity-map solution f, the total-energy function $E(f)$ can be represented as follows:

$$E(f) = E_d(f) + E_s(f) \tag{23}$$

$$E_d(f) = \sum_{p \in P} D_p(f_p) \tag{24}$$

$$E_s(f) = \sum_{p \in P} \sum_{q \in N(p)} C_s(f_p, f_q) \tag{25}$$

where $f_p, f_q \in f$. Also, $D_p(f_p)$ and $C_s(f_p, f_q)$ are defined as follows:

$$D_p(f_p) = |f_p - I_p| \tag{26}$$

$$C_s(f_p, f_q) = |f_p - f_q| \tag{27}$$

In Equations 26 and 27, I_p is an observed value of the disparity value at pixel p of the stereo image. In essence, D_p is a cost function for the data-term of the direct observation of the disparity map, and $C_s(f_p, f_q)$ is a cost function for representing the smoothness by estimating the discontinuities with neighboring pixels. This optimization process is combined with a rapid minimization process called "graph-cut" (GC) [7, 8], in which the best segmentation is achieved through dynamic programing that solves the "Max-Flow Min-Cut Theorem" [61]. The GC method achieves a good local minimum in labeling an image and accomplishes a fast approximation of the disparity map with a well segmented representation and the minimal energy-cost of the disparity map.

3.2.2 Mono-stereo for Proximity Perception

For objects that are too close to get detailed spatial observation with the stereo camera, the monocular camera mounted on the wrist of the mobile manipulator is utilized. Several research efforts exist that employ monocular vision for 3D perception.

For instance, the mono-SLAM with probabilistic 3D map [17] shows good results but requires large computational power for processing the stream of image sequences, which is a common issue in many computer-vision algorithms. To lower the intensity of computation, we propose a process called the "mono-stereo" process for up-close observation of an object with a vision sensor. For this algorithm, the manipulator is controlled to move the monocular camera sideways to capture a pair of images forming a small baseline between the images, and then the stereo matching process discussed in the previous section is reused for disparity map calculation. Post-processing of the disparity image is also required in the case of mono-stereo. The noise issues in disparity calculation that are due to background color and lighting will be addressed in the following sections along with our solution to address this issue.

3.2.3 Post-processing of Disparity Images

To further complete the process of 3D perception through the stereo or mono-stereo disparity-estimation processes, a series of post-processing algorithms are developed to enhance the quality of disparity outputs. First, in the stereo process, a ground plane is estimated using a statistical-region-merging algorithm [56]. This helps to reduce noise and error terms in disparity calculation that are caused by the short distance to the ground or the complexity in the texture of the ground. If properly estimated, this process allows simplification in disparity calculation for non-ground objects that are of interest. Second, also in the stereo process, the histogram distribution of an initial disparity map is analyzed to determine the range characteristics of the environment to generate a mixed-parameter graph-cut disparity output. Lastly, in the mono-stereo process, foreground objects are separated from the background to capture the right partition of the object in the disparity image.

3.2.3.1 Ground Detection

As illustrated in Figure 34. (c) $\sim$ (e), a disparity map alone is insufficient for generating a clean 3D map since it possesses a certain amount of noise and errors resulting from different textures or light reflexes. Therefore, it would be beneficial if the ground plane is properly estimated and eliminated from the disparity map.

Our approach for estimating the ground plane is based on segmentation of the image using texture data. A common segmentation technique that utilizes texture characteristics of the image is the statistical-region-merging (SRM) [56]. To successfully apply this method for ground estimation, first, the color spectrum domain of our stereo images is converted from red-green-blue (RGB) to hue-saturation-value (HSV) to split the spatial distribution and effect from lighting. Then, the SRM algorithm is applied on all three domains to select the common regions from the three estimation pairs, finally resulting in a clean ground estimation as depicted in Figure 35.

3.2.3.2 Histogram-based Mixed-scale Disparity Map

Even after the successful generation of a disparity map and a reliable ground-estimation process, the resulting disparity map may not be detailed enough to describe the spatial distribution of the environment. This is due to the fact that indoor environments typically consist of large variations in the placements of objects that can cause errors in disparity estimation process. Therefore, in the case of a mixed spatial-distribution of objects, we incorporate mixed-parameter sets for the graph-cut algorithm to generate the disparity map.

Thus, the initial disparity map is analyzed by observing the histogram distribution, and the system determines which process to choose from: a single graph-cut parameter set for disparity estimation for distant-spatial perception, a mixed-parameter set for a more detailed disparity over a mixed-spatial environment, or a mono-stereo process for proximity perception. This process is illustrated in Figure 36.

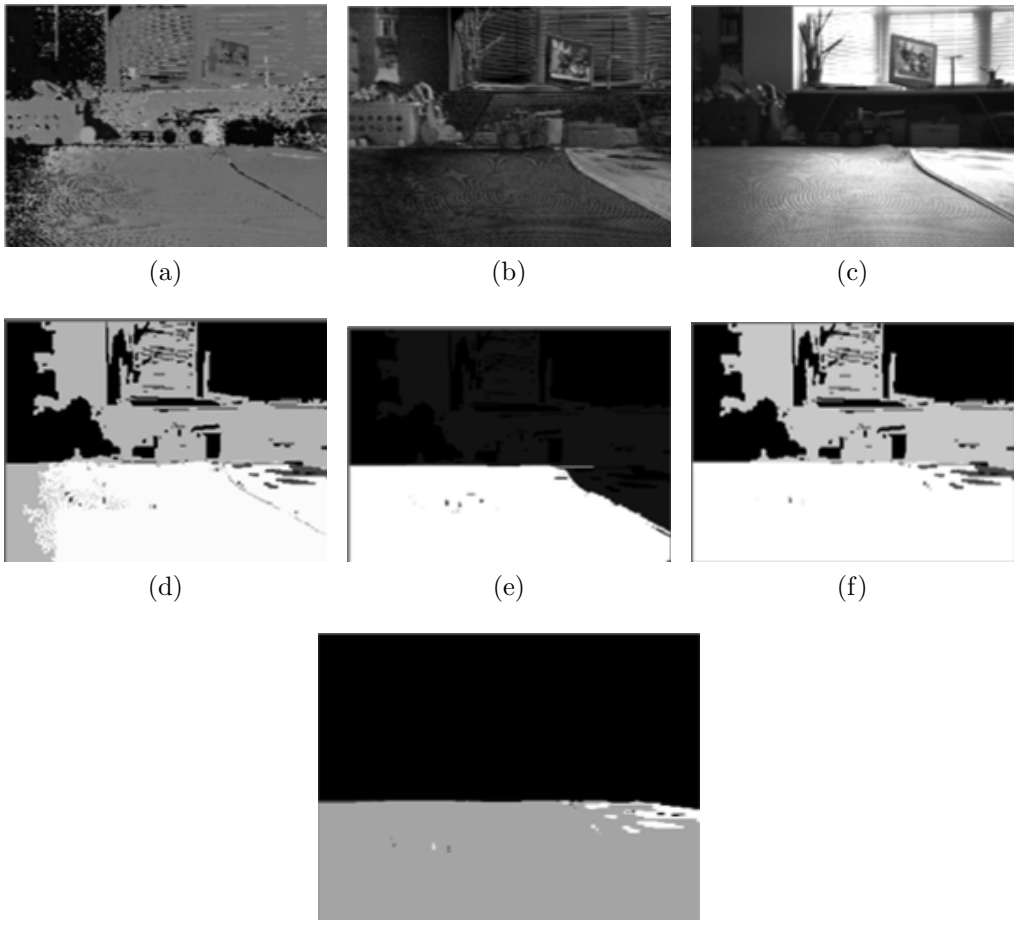


Figure 35: Ground estimation process on hue/saturation/value (HSV) channels using a statistical-region-merging algorithm. (a) Hue channel image of an indoor scene, (b) saturation channel image of the scene, (c) value channel image of the image, (d)–(f) Results SRM process on corresponging (a)–(c) images, (g) final ground estimation from (d)–(f).

3.2.3.3 Separation of Foreground Objects and Background

Unlike the stereo process for distant-spatial recognition, the need for ground estimation disappears in the mono-stereo process. However, new challenges arise in mono-stereo sensing as follows: First, although the camera is observing objects in the front, other trivial objects can be scattered in the background confusing the user during haptic exploration; Second, the disparity calculation fails when the objects are near a white background-for example, a bottle placed in a refrigerator, because

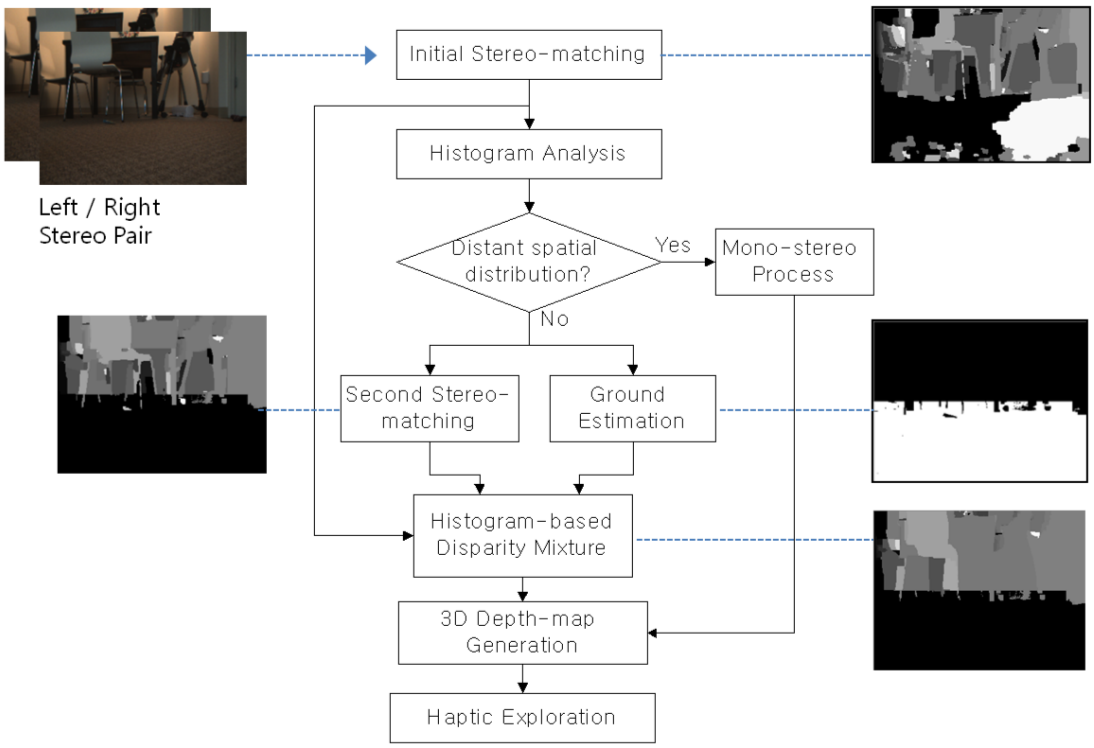


Figure 36: Stereo perception process.

the white walls inside the refrigerator cause the disparity estimation fail on the wall areas by making it hard to find matching points (or features).

The solutions to the above issues, surprisingly enough, come from the inherent nature of the system itself. Since the mono-stereo process employs the movement of the robotic arm for the acquisition of stereo pairs, the process can also observe the variances of optic-flow values of every feature-point on the image domain. Optic flow is the pattern acquired from any motion of objects, surfaces, or edges in visual sensing caused by the relative motion between vision sensor and the scene [32]. Thus, based on the optic-flow distribution, the foreground region (objects) and background region can be differentiated.

To be more specific, we acquire optic flow changes over the pixels in the scene and filters out background pixels with optic flow vectors below a threshold. Then, edge points are sampled to form a set of skeletonized (silhouette-like) point markings

and fed into the SRM segmentation module as seed points to estimate the foreground regions. Finally, we combine the foreground region and disparity results to generate an enhanced disparity estimation for objects at proximity. This process is illustrated in Figure 37, and an exemplary result is shown in Figure 38.

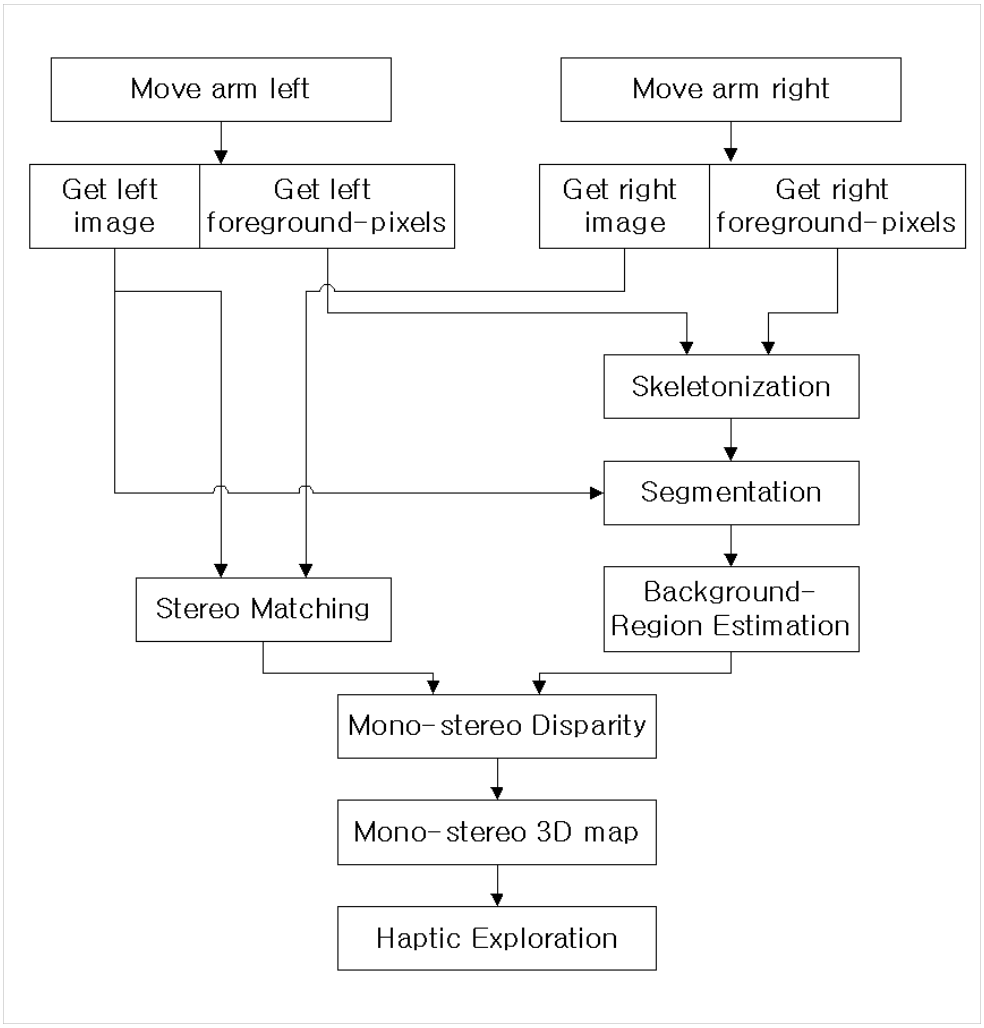


Figure 37: Flow-chart of 3D perception through the mono-stereo process with foreground/background separation.

(a) Left image

(b) Right image

(c) Edges of foreground from left motion

(d) Edges of foreground from right motion

(e) Segmentation

(f) Foreground regions

(g) Initial disparity result

(h) Final disparity result

Figure 38: Example of the mono-stereo process.

3.3 3D Map Building

Once a reliable disparity estimation of the environment is acquired, a 3D map can be built from the disparity map (either from the stereo process or from the mono-stereo process). After the 3D map is acquired, a haptic exploration process is then activated that interactively searches the map with the movement of the haptic probe and calculates haptic forces in real time.

The usual process for 3D map generation from a disparity map involves a projection matrix built from camera parameters. However, the stereo matching process used in this research includes a rectification process that removes some of the distortion in the acquired images caused by the characteristics of lenses. Thus, a direct convolution with the projection matrix based on a pin-hole model rather distorts the 3D environment again.

Therefore, the disparity data is processed in a different way. Instead of projecting the 2D disparity outward to the 3D spatial map (forward projection), the 2D disparity map is placed at the maximum-depth location and projected inward to form a 3D map (backward projection). The maximum depth can be defined since the process deals wth an indoor environment. This backward projection makes the calculation more efficient and faster than the forward projection, since the bound of the volume data is known a priori. Also, instead of the projection matrix that is based on the camera parameters, a new triangulation process is formulated. The details of the projection for the stereo disparity map are illustrated in Figure 39 and Eqs. 28 and 29. For a 3D position $R(Rx, Ry, Rz)$ that is projected onto a 2D point $I(Ix, Iy)$ on the image plane, the correlation between points can be defined as follows:

$$l_{stereo} + R_z : l_{stereo} + D_{Max} = X_{offset} - R_x : X_{offset} - I_x \tag{28}$$

$$l_{stereo} + R_z : l_{stereo} + D_{Max} = Y_{offset} - R_y : Y_{offset} - I_y \tag{29}$$

where l_{stereo} is the distance from the camera baseline to the projection point

(Origin) for the 3D mapping, D_{Max} is the maximum depth estimable by the stereo camera, and X_{offset} and Y_{offset} are the center coordinates of the image frame (For example, if the image resolution is 320x240, $X_{offset} = 160$ and $Y_{offset} = 120$.)

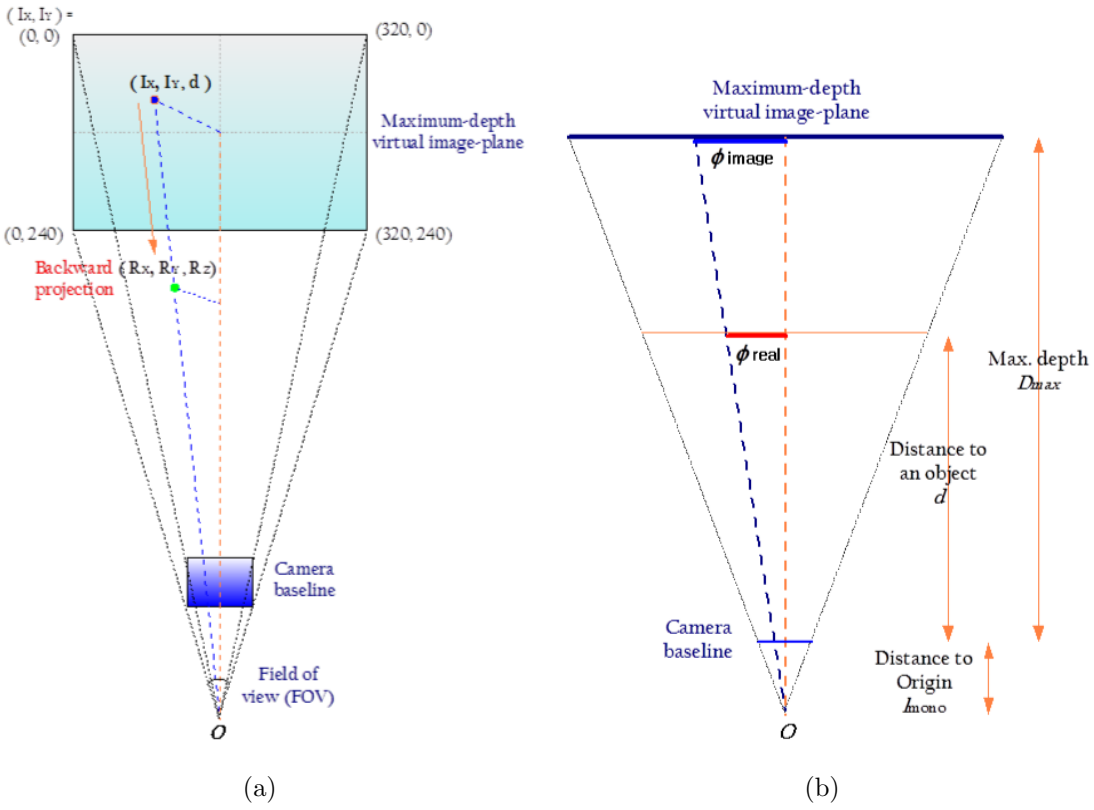


Figure 39: Backward 3D projection for stereo disparity image.

For mono-stereo matching, the movements of the monocular camera enable a stereo observation of a scene at closer distance, which form a smaller baseline and a narrower field of view than the stereo process, which in turn places the origin at a further distance as illustrated in Figure 40.

3.4 Haptic Rendering from Vision-based Robotic Perception

Haptic rendering [75] involves 1) calculating haptic forces arising between virtual objects and a haptic probe and 2) generating the force on a haptic interface so the human user can physically feel the interaction in the virtual environment. Since it

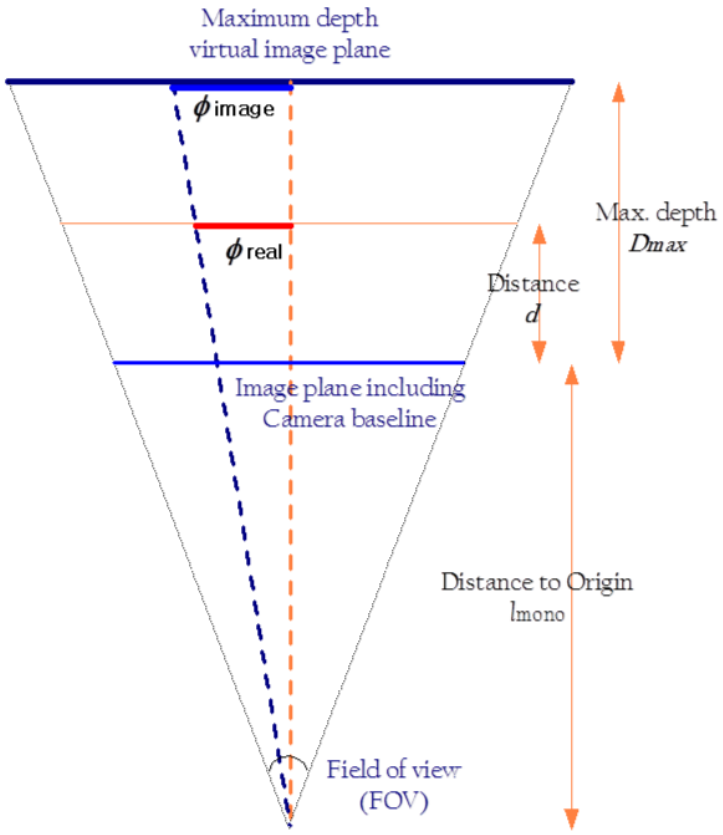


Figure 40: Backward 3D projection for mono-stereo disparity image.

takes significant amount of computing power to handle virtual objects in a virtual environment and to calculate the forces from virtual interactions, the generation of models for the virtual objects and the environment has been usually processed *a priori*. As such, haptic rendering of real-world environments has not been challenged thoroughly. However, it is imperative for the telepresence technology for visually impaired users to have methods for representation of real-world environments through non-visual means. This motivates us to pursue the challenging goal of haptic rendering of real-world objects and environments directly from robotic visual perception.

To enable direct haptic rendering from the robotic sensory data itself, we discuss a totally new approach, which is one of the key contributions of this work. In typical haptic rendering applications, haptic forces are created from a 3D graphical model in a virtual environment. Instead of creating a 3D graphical model, this proposed

research uses point-cloud based 3D data and directly applies a virtual proxy algorithm for haptic rendering. The disadvantage of this approach is that performance is limited by the accuracy of the 3D estimation process, i.e. it cannot represent what it cannot perceive. However, this also applies to other conventional methods in that haptic rendering cannot out-perform the 3D model's representation. The advantages, on the other hand, are fast computation time and capacity for extra sensory modalities, i.e. this method can even encapsulate sound data or dynamic movements in real-time that are not possible with any graphically reconstructed models.

3.4.1 Haptic 3D-Rendering Algorithm

A widely used technique for haptic rendering is the virtual-proxy algorithm [72]. In essence, the virtual-proxy method generates force feedback based on positional differences and velocity differences between a user's haptic probe and the virtual proxy (the representation of the local data of an object's volume at the point of touching). The first step in the virtual-proxy method is to define a volumetric or textural representation of an object at the point of contact. This representation, known as the virtual proxy, is as denoted in Equation 30 and illustrated in Figure 41.

$$\overrightarrow{f}_{feedback} = k\left(\overrightarrow{P}_{proxy} - \overrightarrow{P}_{probe}\right) + D\left(\overrightarrow{v}_{proxy} - \overrightarrow{v}_{probe}\right) \tag{30}$$

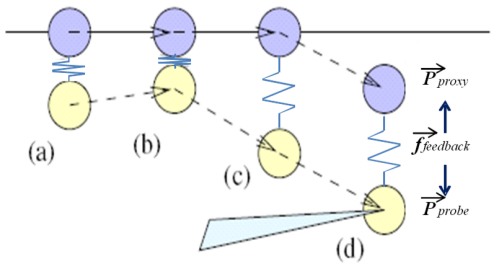


Figure 41: Virtual-proxy method for haptic volume representation.

Given the position vector of the proxy $\overrightarrow{P}_{proxy}$, a position vector of the probe

$\overrightarrow{P}_{probe}$, and the velocities of the proxy and the probe, $\overrightarrow{v}_{proxy}$ and $\overrightarrow{v}_{probe}$, a virtual-proxy force-feedback $\overrightarrow{f}_{feedback}$ is composed of a penetration depth term (with a spring constant k) and a damping term (with a damping constant D) as shown in Equation 30. The penetration depth term is a static value representing the positional relationship between $\overrightarrow{P}_{probe}$ and $\overrightarrow{P}_{proxy}$ that conveys object shape and stiffness to the user. The damping term is a dynamic value that corresponds to the movement of the probe. The corresponding proxy position represents the surface friction or texture of the object.

By probing the space with the haptic device, the user can now perceive the 3D representation of the object shape. Figure 42 shows the 3D map of a bottle with occupied grid-points representing the object and unoccupied points representing empty space. The virtual proxy force to be exerted upon the physical haptic interface is calculated by finding the shortest vector to the surface of the virtual representation of the bottle from the virtual probe inside the occupied points. More precise forces for representing the detail of the surface is achieved by estimating the surface proxy force using predefined surface normal vectors corresponding to the occupancy of neighboring grid-points as illustrated in Figure 43.

3.4.2 Haptic Interaction with Robotic Sensor Data

Once the 3D map is constructed from the fusion of sensory perception, the system can then enable a user with a haptic device—the Phantom Omni from SensAble Technologies—to explore the 3D space and feel objects in it using the virtual proxy algorithm [47]. An interesting problem, however, arise from the sparseness of the grid-based 3D map. Although the 3D map consists of millions of grid points, it is still a sparse data-set in comparison to the large volume encompassed by the spatial environment. Therefore, we implement a 3D-interpolation process for smooth representation of surfaces along with a proxy-based penetration calculation algorithm.

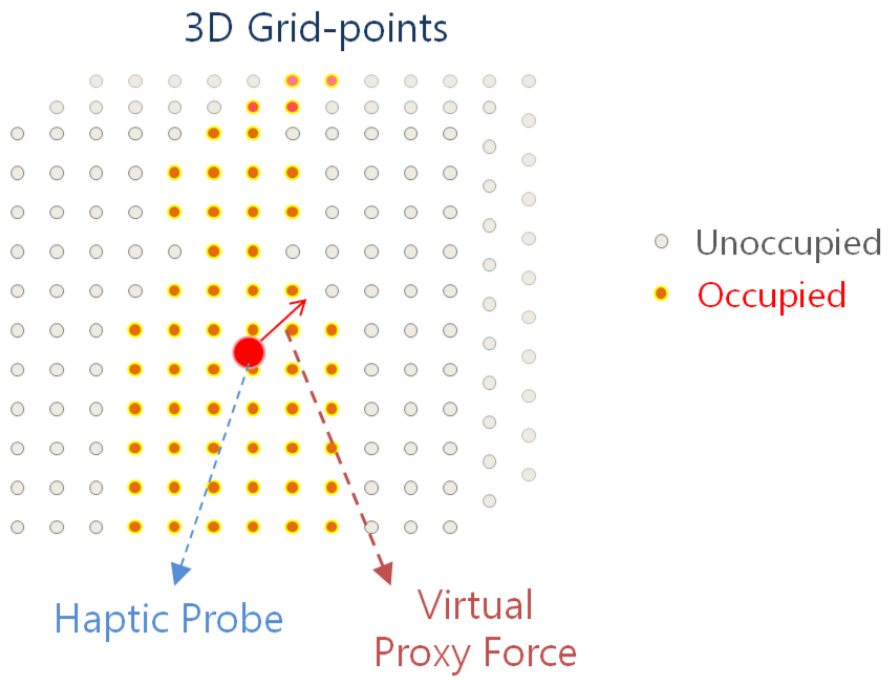


Figure 42: Virtual proxy force in a 3D map representing a bottle.

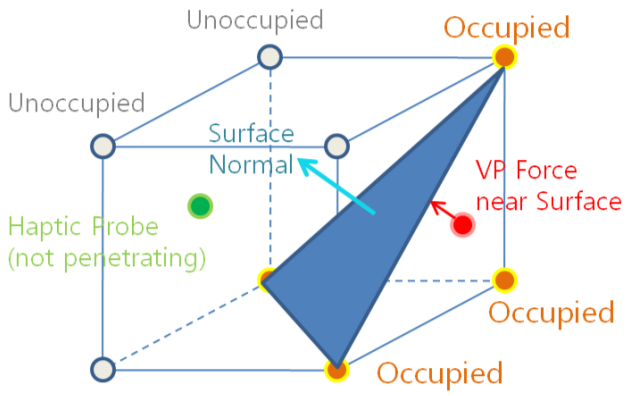


Figure 43: 3D Point-Map Search for Haptic Exploration.

This research mainly focuses on surface representation, so the virtual-proxy is placed on the surface of a volumetric object. The distance between the virtual-proxy placed on a surface and the position of a probe (Omni's end-point) is used to calculate the penetration depth of the probe with respect to a surface point (a virtual-proxy). This parameter is then used to generate the force feedback for haptic rendering. Since a grid-based sparse 3D data-set is being used, the haptic interaction algorithm first checks for neighboring grid-points' occupancy with respect to the position of the probe of the Omni in the 3D map. Then, the algorithm calculates the penetration depth if the probe is totally inside the object's volume or calculates surface penetration depth if the probe is near the surface, as described with a flowchart in Figure 44.

The volume proxy-force is calculated by searching for the closest surface to get the distance vector. The surface-proxy force is calculated by the convolution of the surface-normal vector and the penetration depth from the surface, which leads to an interpolation issue caused by the data sparseness. The processing time for the 3D-map search process and proxy-force generation is crucial for a real-time experience of the haptic exploration, and the implementation of the algorithms explained above results in an average search-time of 3.075ms (with the standard-deviation (STD) of 6.287ms over 7,035 measurements) for surface-normal depth calculation in the 3D grid-map and proxy-force generation process.

3.5 Results

The haptic exploration methodology, through a human-user study was performed to evaluate the performance of the system working in real indoor-environments. For evaluating the processes, a divide-and-conquer approach was employed. The methodology was evaluated based on the results from the three major sub-processes: stereo matching with ground elimination, mono-stereo matching with background elimination, and the haptic exploration process. For the first two matching processes, a

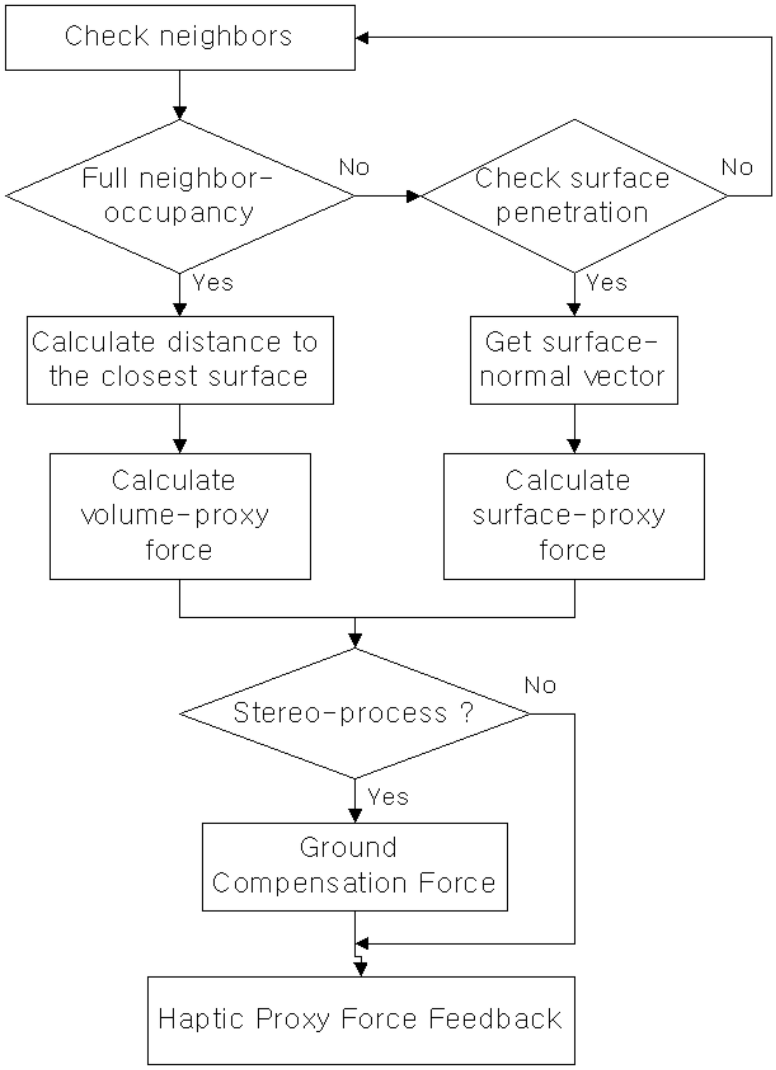


Figure 44: Haptic virtual-proxy algorithm for calculating interaction force with a 3D grid-map.

ground-truth map of the real world environment was manually generated, and compared with the algorithm output. For evaluating the haptic exploration process, several different objects from real-world observation were chosen, and human-users (both sighted and visually impaired) were allowed to explore the 3D map of the object with the haptic probe, without seeing the object. After the randomized exploration, the subjects were asked to tell or choose which objects they were touching (from the images of objects or the set of actual objects).

Table 3: Processing time and RMSE value of depth error for image pairs in Figure 45

Stereo Pairs	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6
Processing Time (sec)	15.1	17.6	21.7	20.9	14.4	14.9
Depth Error (%)	4.53	16.1	–	–	–	–

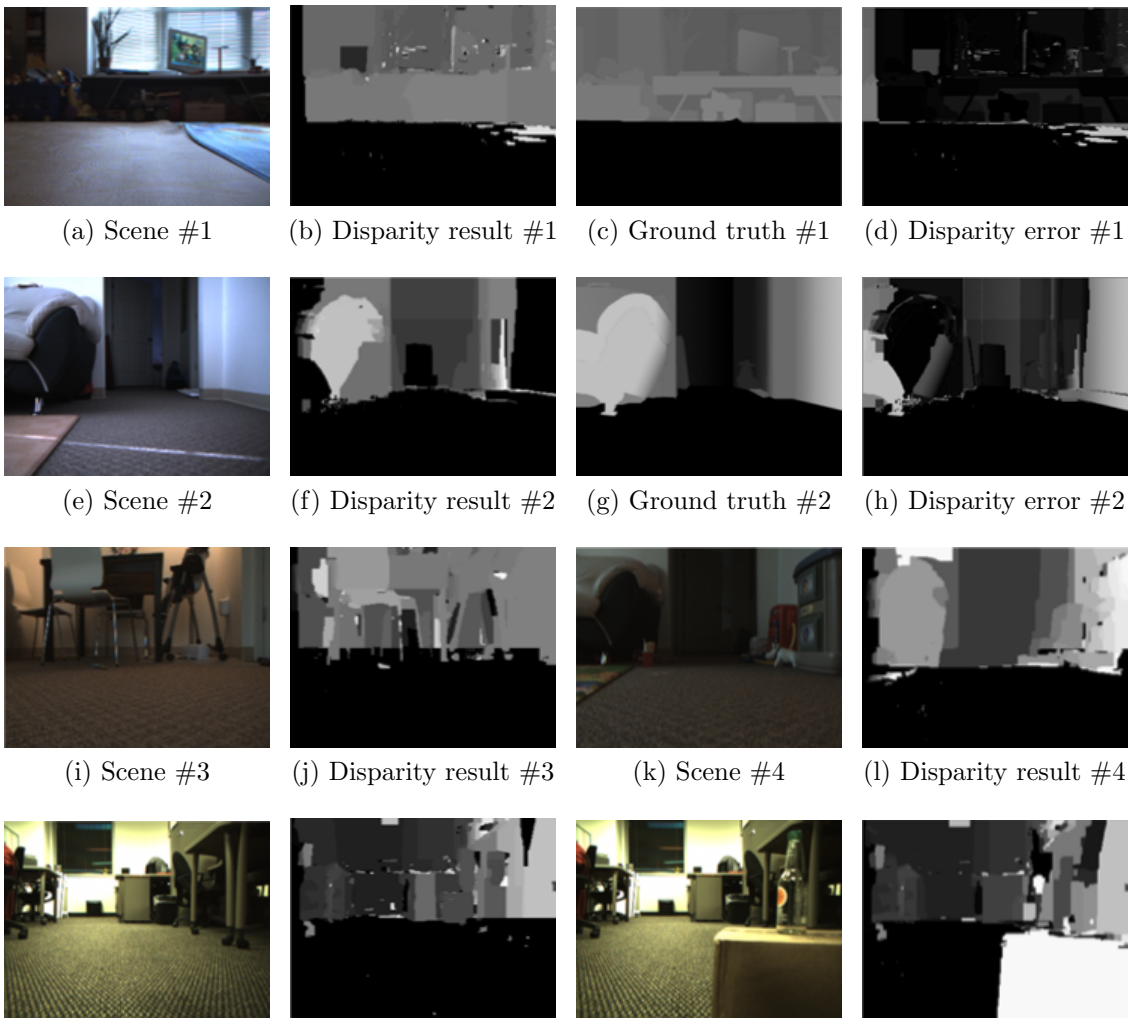
3.5.1 Stereo Process Results

Figure 45 presents the results of the stereo-matching algorithm with ground elimination and the corresponding ground-truth maps. Four different scenes were taken from an actual household living environment, and two scenes were taken from a real office environment. Two sets of ground truth data were measured over the 3D environment and then translated into disparity maps. For equal comparison, the ground planes were also eliminated from the ground truth map with manual selection. The absolute difference of disparity values over the entire image plane was calculated and the root-mean-squared error was computed.

As can be seen from Figure 45 and Table 3, the average processing time for the stereo matching and 3D map generation was 17.4sec, and the errors between the algorithm's estimation and the ground truth were 4.53% and 16.1% respectively. The higher error rate in Scene #2 is due to the featureless regions, the dark bottom-side of the sofa and the plain white wall. However, those areas with disparity estimation failures were scattered toward the outside boundaries of the environment, our process works fine with different lighting and more cluttered environment as in Scene #4.

3.5.2 Mono-Stereo Process Results

Figure 46 displays results from the mono-stereo matching algorithm with background elimination, along with the corresponding ground-truth maps. Six different objects were placed randomly in a small refrigerator, the door of the refrigerator, or on a table.



(a) Scene #1 (b) Disparity result #1 (c) Ground truth #1 (d) Disparity error #1

(e) Scene #2 (f) Disparity result #2 (g) Ground truth #2 (h) Disparity error #2

(i) Scene #3 (j) Disparity result #3 (k) Scene #4 (l) Disparity result #4

(m) Scene #5 (n) Disparity result #5 (o) Scene #6 (p) Disparity result #6

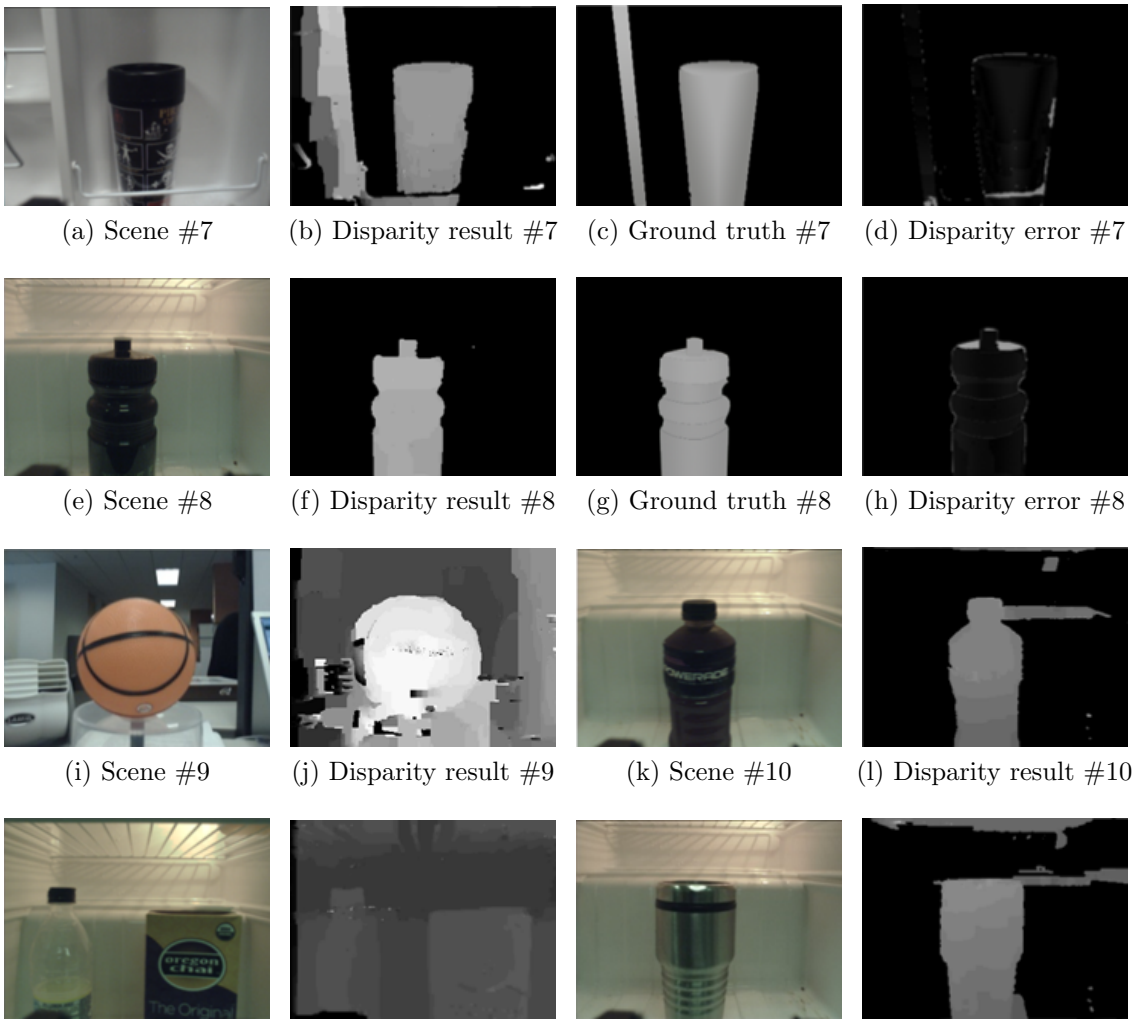
Figure 45: Results of stereo matching process in real environment.

The disparity map was generated from the mono-stereo algorithm utilizing a moving robotic arm, and the background was eliminated if the number of segments detected in the image reached a threshold. Two sets of ground truth data were measured over the environment and then translated into disparity maps. For equal comparison, the background was also eliminated from the ground truth map with manual selection. Unlike the stereo matching process, since only the partial map of the ground truth is given, the error is calculated based on the provided area of the ground truth. Table 4 shows that the average processing time for the mono-stereo matching and 3D map

Table 4: Processing time and RMSE value of depth error for image pairs in Figure 46

Stereo Pairs	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6
Processing Time (sec)	6.85	6.5	6.7	7.06	6.1	6.8
Depth Error (%)	4.6	7.9	–	–	–	–

generation was 6.7 sec and the errors between the algorithm's estimation and the ground truth were 4.6% and 7.9%, showing the accuracy of the process.



(a) Scene #7 (b) Disparity result #7 (c) Ground truth #7 (d) Disparity error #7

(e) Scene #8 (f) Disparity result #8 (g) Ground truth #8 (h) Disparity error #8

(i) Scene #9 (j) Disparity result #9 (k) Scene #10 (l) Disparity result #10

(m) Scene #11 (n) Disparity result #11 (o) Scene #12 (p) Disparity result #12

Figure 46: MonoStereo process results.

3.5.3 Human Subject Results with Haptic Rendering

Once the stereo matching or mono-stereo matching process are complete and the 3D map is generated, haptic exploration is activated and a human user is allowed to feel the environment with the PHANToM Omni. For validation, three objects were selected from results of the mono-stereo matching process (stereo matching results were not used due to the small workspace of the Omni) and human users were provided with three sets of the haptic model based on the objects shown in Figure 47. The users were provided with a randomly selected sequence of objects represented in haptic space, and were asked to match with the list of objects after a minute of observation time. For subjects with visual impairments, the three objects were provided so they could sense the real objects. For sighted subjects, considering the fact that they feel more comfortable in identifying objects based on visual perception than haptic perception, a paper with the images of the three objects are provided so they could point out the matching object. A total of 19 sighted human subjects and eight subjects with visual impairments participated in the experiment. Of the participants with visual impairments, seven were partially blind and one was fully blind.

(a) (b) (c)

Figure 47: Images of objects of which the haptically visualized model is provided for haptic exploration for human users.

The results are shown in Tables 5 and 6. Subjects with no visual impairments showed 79% to 84% recognition rate for the three objects, while subjects with visual impairment rated 87.5% for each objects. To understand which objects confused the

Table 5: Results of haptic object-recognition experiment with subjects without VI after mono-stereo process

	Object 1	Object 2	Object 3
Success Rate	84.2%	78.9%	84.2%

Table 6: Results of haptic object-recognition experiment with subjects with VI after mono-stereo process

	Object 1	Object 2	Object 3
Success Rate	87.5%	87.5%	87.5%

human users, the confusion matrices for the responses of the participants are listed in Tables 6 and 7. Table 7 reveals that two non-VI subjects became confused between object 2 and object 3 and two other non-VI students had trouble distinguishing object 1 and object 3, mainly because the objects are commonly up-right shaped. For VI subjects, Table 8 shows that one subject became confused between object 2 and object 3, and another subject could not tell the difference between object 1 and object 3, the reason being the same as in the non-VI subjects' case.

To illustrate the force profile the user was feeling and the force amplitude, Figure 48 shows the partial force trajectory while a user touches the side of an object (Figure 47-(b)).

3.6 Conclusions

As shown through the above sections, the proposed HEMM system was able to generate a 3D environmental model using robotic visual sensors with a small processing time (in the order of seconds) and represent this model to the human user in real-time using haptic perception. Human subjects were able to feel a remote object using the HEMM system and were also able to distinguish objects using only haptic exploration.

Table 7: Confusion matrix of subjects without VI in identifying objects with haptic exploration

Object / Answer	1	2	3
1	16	-	1
2	1	15	2
3	2	4	16

Table 8: Confusion matrix of subjects with VI in identifying objects with haptic exploration

Object / Answer	1	2	3
1	7	-	-
2	1	7	1
3	-	1	7

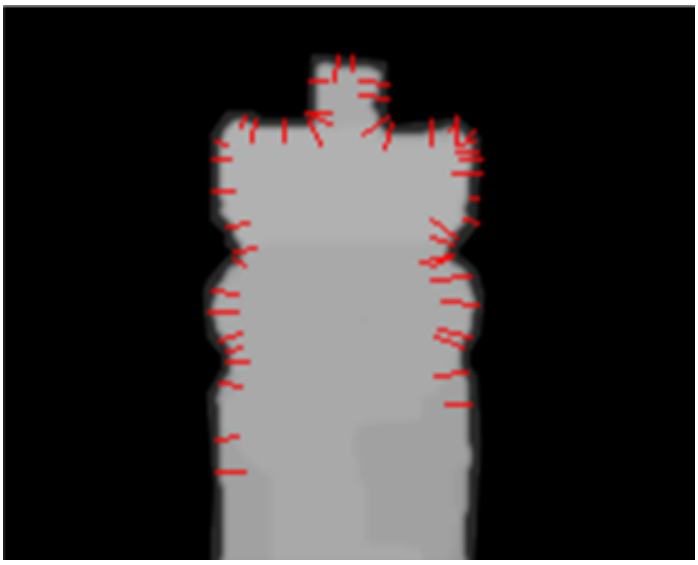


Figure 48: Force trajectories during haptic exploration of an object.

The next step of this research, then, is to incorporate more active visual sensors, such as depth cameras, to design a real-time haptic exploration and telepresence system.

CHAPTER IV

HAPTIC TELEPRESENCE AND MULTI-MODAL INTERACTION

Given direct 3D haptic exploration from robotic visual perception, we now tackle the challenging issue of developing a telepresence robotic system for individuals with visual impairments. For this purpose, the Kinect sensor is utilized to provide active depth sensing with high update ratio of 30 fps for raw sensor data, and a microphone array for adding auditory perception. With this additional sensory device, the objective of this chapter is to describe our system architecture for a telepresence robotic system. Using the same haptic device, this telepresence robot can simultaneously model a remote environment in a 3D haptic space as well as be controlled by the user. Through this novel system design, the goal of this research is to extend the boundaries for activities of daily living for individuals with visual impairments. For example, an application of this technology can be the ability to command a robot from one's study room to clean certain messes on the living room floor, or to command a robot to find a coke bottle and bring it to the user.

The details of this research effort are explained as follows: Section 4.1 describes the overall system architecture, and Section 4.2 details system blocks and functional modules of the system. Section 4.2.1 deals with the real-time update issues for grabbing depth information from the 3D environment, followed by Section 4.2.2, which discusses temporal filtering for smooth haptic rendering and visual representation for enhancing the user's experience. Section 4.2.3 explains the approach for multimodal feedback for the user designed to aid in the user's in-situ knowledge of the remote

environment and the robot. Section 4.2.4 details the framework to integrate the functional capabilities of the system and enable a real-time experience of the system in a real-world environment. Section 4.3 explains the experimental setup and scenarios for evaluating the system, and Section 4.4 shows the results with analysis and discussion.

4.1 System Architecture for a Telepresence Robotic System

Figure 49 visualizes the whole system architecture and functional modules of the haptic telepresence robotic system developed in this research effort. The main hardware platform is composed of a robotic arm (Pioneer2 Arm) and a robotic mobile base (Pioneer 3AT), equipped with a stereo camera, monocular camera, and the Kinect depth sensor with an embedded microphone array. The system interfaces with a user in three ways: 1) visual feedback through a graphic-user-interface (GUI) for typical users, 2) haptic feedback through a haptic interface for visually-impaired users, and 3) verbal feedback through a speaker or headset for visually-impaired users.

The main control architecture is composed of four major functional blocks: 1) vision processing module, 2) active depth/audio perception processing module, 3) haptic interaction module, and 4) mobile manipulator control module. The vision processing module is as described in the previous Section 3.2 and is not discussed in this chapter. The haptic interaction module is an extension of the virtual-proxy based haptic interaction module described in Section 3.4 that encompasses real-time updates from Kinect, and the handling of multiple haptic feedback events. As such, the haptic interaction module works under the supervision of a finite-state-machine (FSM) that governs state changes and submodule controls per states. The last module, the mobile manipulator control module, includes the FSM and sub-control blocks for robotic arm kinematics calculation and control as well as robot localization and navigation control of the mobile base. One last addition is the verbal feedback block, which works synchronoulsy with the haptic interface, 3D map, FSM, and robot navigation

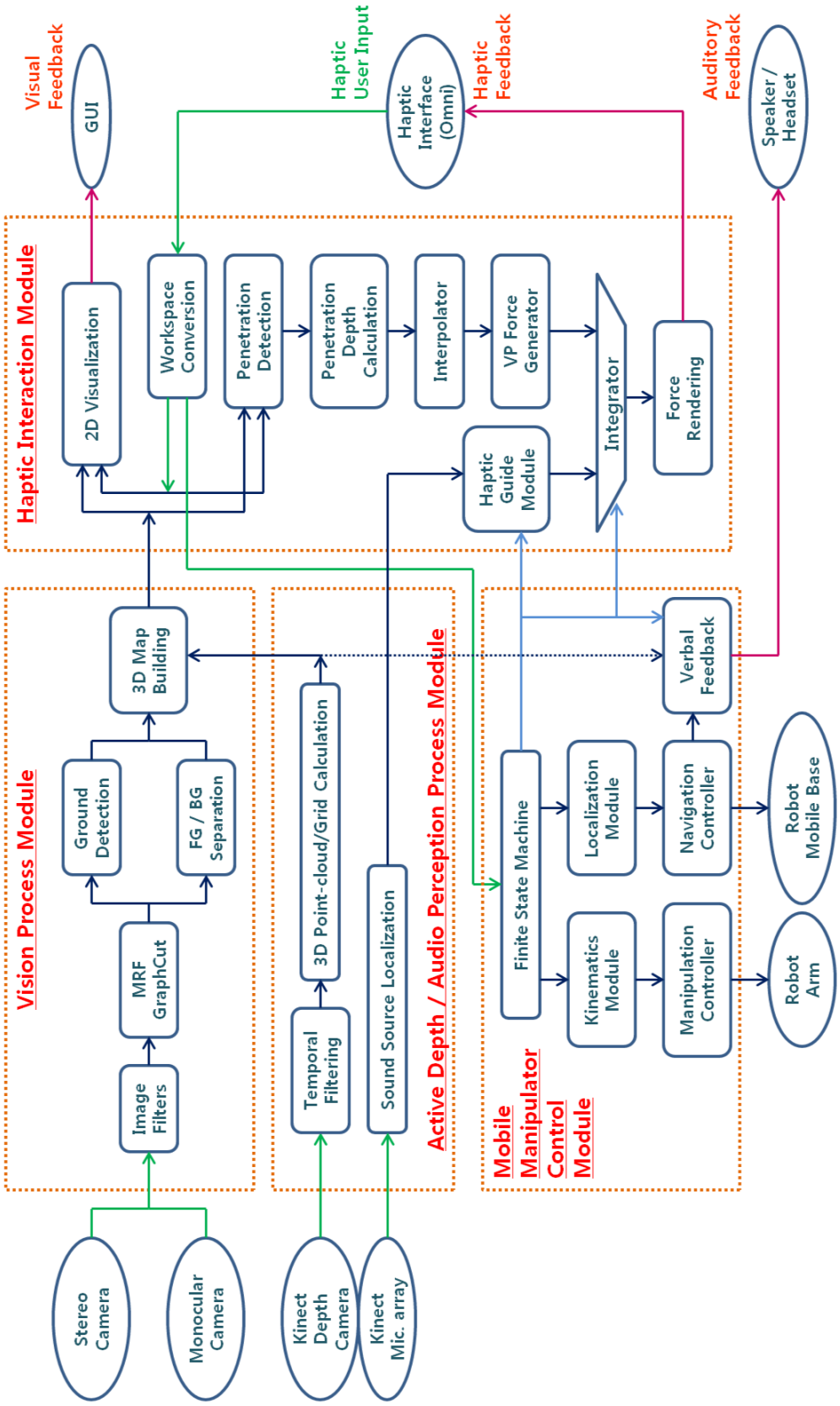


Figure 49: Haptic telepresence robotic system archicture.

controller to present verbal feedback to the user. The details are explained in the following section.

4.2 System Blocks and Functional Modules

The four major functional blocks, mentioned in Section 4.1, are constructed to work synchronously to achieve the objectives of our system. This section explains the functional blocks of the system, which are the Kinect-based active depth-sensing and audio sensor module, the haptic interaction module, and the mobile manipulator control module.

4.2.1 Handling of the Information Flow from the RGB-D Sensor

The Kinect sensor from Microsoft is an RGB-D sensor that provides depth and color images at a rate of approximately 30 frames per second (fps). However, the processing time for building a 3D map based on a single sequence of depth and color data takes more than 100ms on a desktop setting with Intel i7 core and 6GB flash memory. Therefore, to handle continuous depth and color image input-flow from the Kinect, a two-stage pipeline structure has been constructed as described in Figure 50. In the first stage, the Kinect depth data frame is buffered, projected into 3D coordinates for each 320x240 data points, transformed into a 3D grid map, translated into a disparity map for visualization, and calibrated for distance in case of scaling. The corresponding color image is buffered at the same time and is used to provide color information at locations connected with haptic probing. The second stage of the pipeline handles the haptic interaction, which constantly accesses the 3D map to calculate virtual proxy forces for the haptic probe while updating the 3D map if a new map is built. The update process only requires swapping of the memory address, so it does not affect the real-time process of the haptic interaction that is currently handled approximately at 300 Hz.

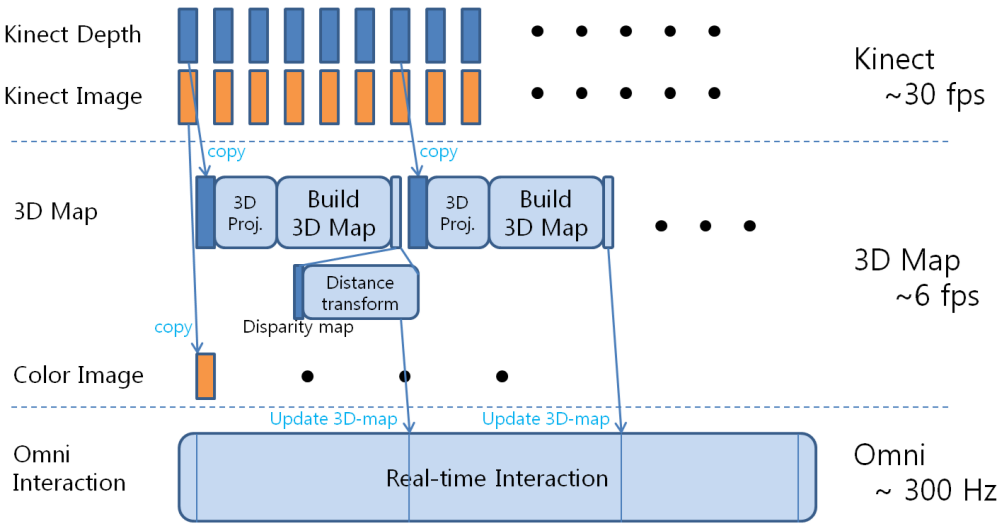


Figure 50: Kinect update flow for 3D map building and haptic interaction.

4.2.2 Handling of the Sensory Limitations of the RGB-D Sensor

The Kinect sensor, despite its outstanding performances and cost-efficiency, has two characteristics that limits the quality of its sensory perception. First, since the depth is measured by the array of reflected lights emitted from a small source emitter (Figure 51), the reconstructed 3D points inevitably contains large shading areas that cannot be detected (Figure 52b). Secondly, the inherent noise in the sensor data creates random dark spots and broken edges, which can cause cracks and rough surfaces in the constructed 3D map.

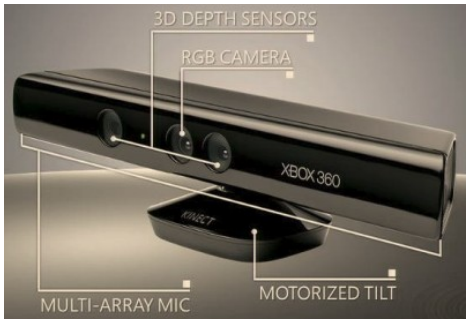
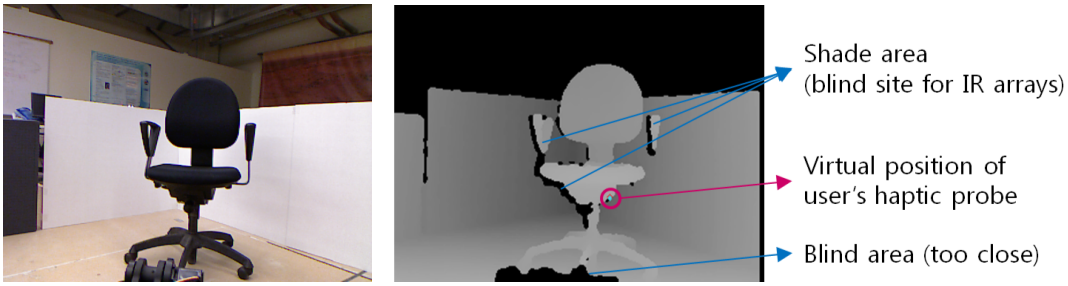


Figure 51: Kinect module.

Since we seek fast algorithms to enable real-time implementation, we provide a solution to the above issues by using temporal filters. As Figure 53 illustrates, the



(a) Color image view (b) Depth view, shades, and the user's probe

Figure 52: Haptic probing of a chair in a 3D map.

noisy kinect depth data goes through a temporal filtering process to reduce the effects from the two issues. The temporal filter first finds the shade and fills the shade with distance-based smoothing, then peforms pixel-wise median filtering to reduce outliers ("dark spots") due to the sensory noise. Figure 54 shows the difference between the raw depth data and the filtered depth data.

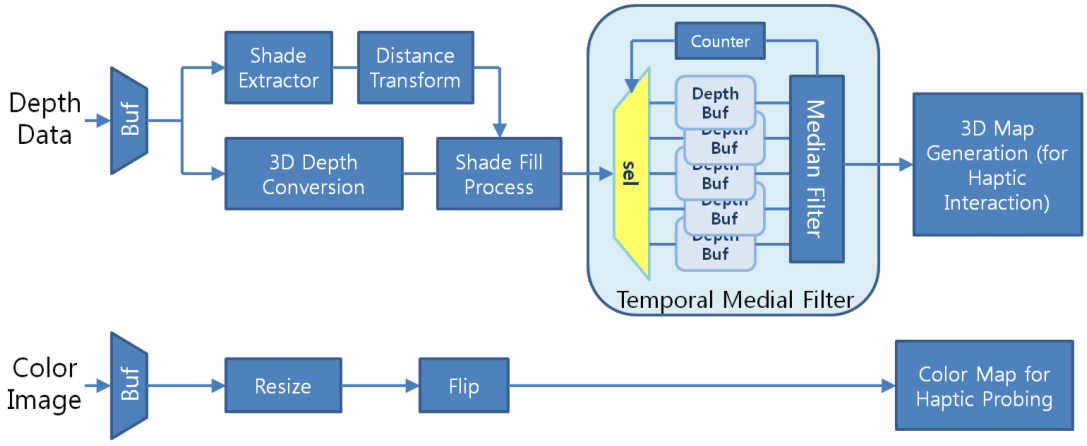
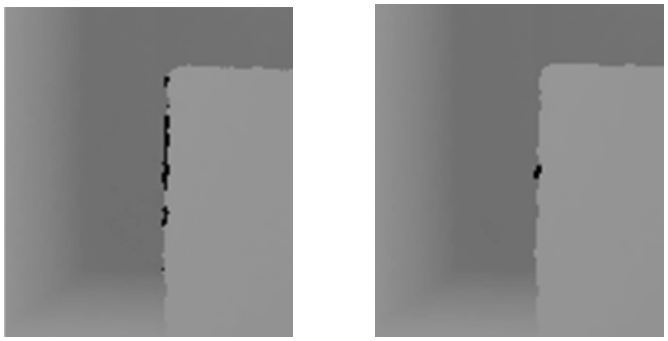


Figure 53: Temporal filtering of Kinect sensor data.

4.2.3 Multi-modal Feedback for Tele-perception

To provide in-situ perception for a remote environment in non-visual ways, we utilize haptic and auditory feedback. Haptic force feedack is used to transfer information to the user through the user's tactile perceptioin, and we encode three kinds of information into the haptic channel: object shape, guidance for the user, and notification



(a) Before filtering (b) After filtering

Figure 54: Result of the temporal filtering.

of an event. Auditory feedback is intended to provide information that cannot be transferred through the haptic channel, such as the color of an object or the status of the robot.

4.2.3.1 Haptic Feedback

Figure 55 describes the three different haptic feedback cues that are integrated for the user. The haptic feedback sources can be selectively activated to transfer appropriate force-feedback to the user. The Virtual Proxy Force explained in Section 3.4 is the first haptic feedback source that is formed by the user's haptic exploration in the 3D space mapped from robotic perception.

The Guiding Force can function as an input to the user, i.e. a event message with haptic modality. Specific examples include notifying robot's status or environmental events (such as sensing of loud noise in a remote place) with a haptic vibration or pulling force feedback as discussed in Section 2.2.

The Navigation Control Force can assist the user in controlling the robot by constructing haptic-support-planes (HSP), a set of force walls, as illustrated in Figure 56. The user can move the probe (and the robot) in the intended direction, and a haptic force will be generated to keep the probe inside the HSP space. In this way, the haptic device becomes a 3D joystick with force-field support. The advantages of this method is that the user can feel more exactly which directional command is

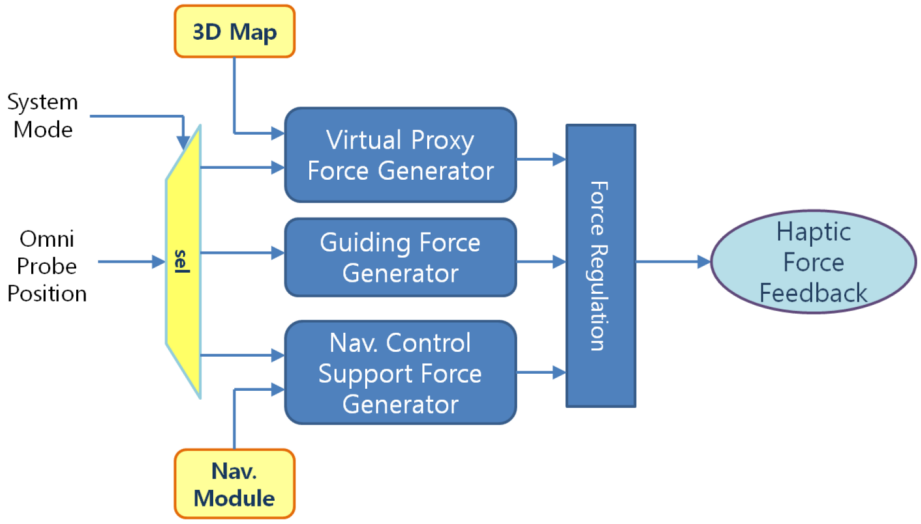


Figure 55: Haptic force-feedback integration.

being given, while still being able to set precise commands for the heading and speed of the robot.

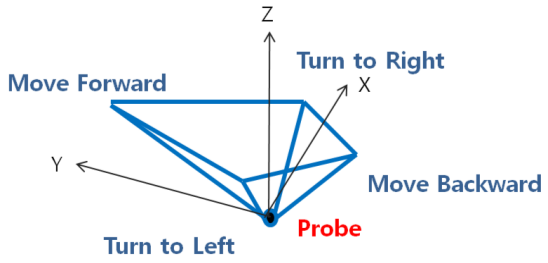


Figure 56: Haptic-support-planes for assisting navigation control.

4.2.3.2 Auditory Feedback

The auditory feedback consists of verbal descriptions of color and distance information assiciated with an object the haptic probe is in contact with. It also consists of a brief verbal report on the status of robotic movements. The color and distance information is reported to the user when the haptic probe is interacting with an object in the 3D map (in other words, "penetrating" the virtual objects in the 3D map). The list of colors recognized by our system is shown in Table 9. The verbal description of the status of our robotic system consist of "forward", "left", "right", "backward", and

Table 9: List of color names differentiable with verbal feedback.

#	Color	#	Color	#	Color	#	Color
1	Aqua	18	Gold	35	Midnight blue	52	Sea green
2	Aquamarine	19	Green	36	Navy blue	53	Sienna
3	Black	20	Green yellow	37	Neon blue	54	Sky blue
4	Blue	21	Grey	38	Olive	55	Slate blue
5	Blue violet	22	Hot pink	39	Olive green	56	Spring green
6	Bronze	23	Indian red	40	Orange	57	Steel blue
7	Brown	24	Ivory	41	Orange red	58	Summer sky
8	Cadet blue	25	Khaki	42	Orchid	59	Turquoise
9	Charteuse	26	Lawn green	43	Pale green	60	Violet
10	Chocolate	27	Light blue	44	Pale violet	61	Violet red
11	Coral	28	Light coral	45	Peach	62	White
12	Dark green	29	Light gold	46	Pink	63	Wood
13	Dark grey	30	Light sky blue	47	Plum	64	Yellow
14	Deep pink	31	Lime green	48	Purple	65	Yellow green
15	Deep sky blue	32	Magenta	49	Red		
16	Dodger blue	33	Maroon	50	Royal blue		
17	Forest green	34	Medium purple	51	Scarlet		

"stop", and are reported only when the status changes. i.e. it does not keep reporting "forward" while moving forward but reports only once.

4.2.4 Finite State Machine

To integrate all functionalities into a unified system, we constructed a finite state machine (FSM) to run seemlessly while interacting with the human user (Figure 57). The FSM is composed of five states with eight events that trigger the transition between the states. Each state define sub-states for the robotic platform and the

haptic interface. The whole FSM configuration is governed by three scenarios designed to encapsulate the experimental setups used to validate the system functionality and evaluate the three hypothesis later described in Section 4.3. These three scenarios are described as below:

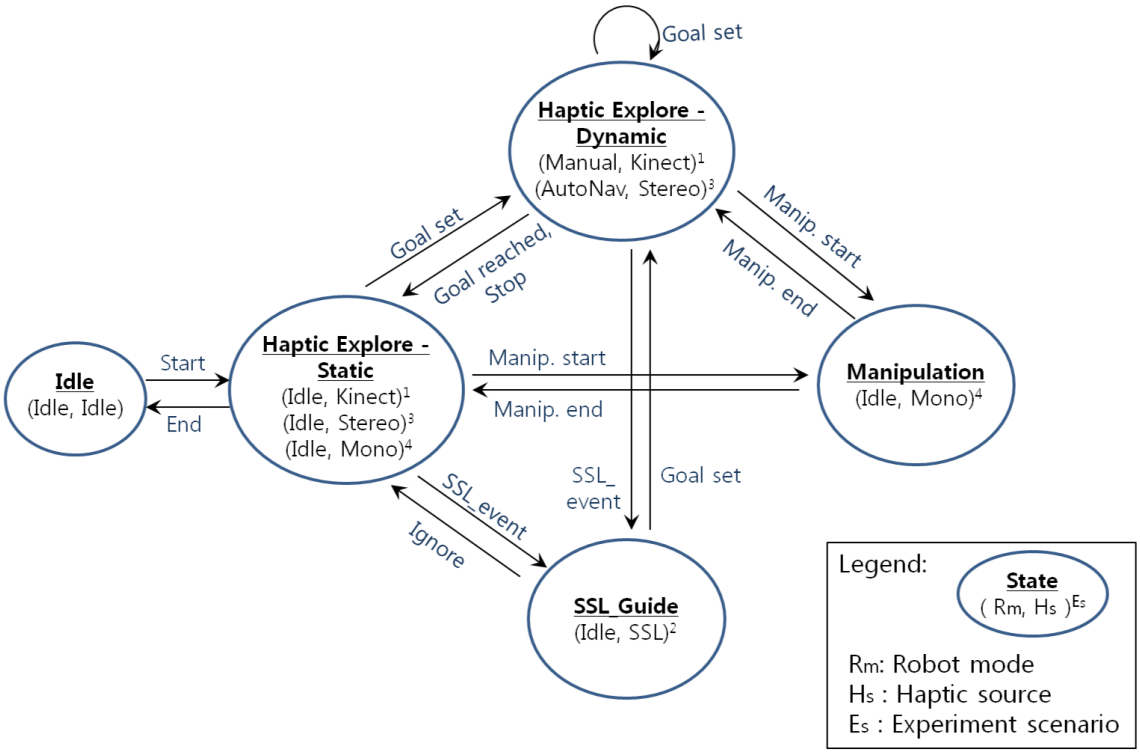


Figure 57: Finite-state machine (FSM) for our haptic telepresence system.

- Scenario 1: Keyboard control with multi-modal tele-perception. The human subject holds the haptic interface to feel and explore the spatial distribution of the remote scene, and uses the traditional keyboard control of "up arrow" for making the robot move forward, "down arrow" for making the robot move backward, "left arrow" for making the robot turn left, "right arrow" for making the robot turn right, and "Ctrl key" for making the robot stop.

- Scenario 2: Haptic-support-plane control multi-modal tele-perception. The human subject holds the haptic interface, which is confined to a force field as

depicted in Figure 57, and controls the robot to move toward the intended direction. Compared to Keyboard control which can only give four directional controls, this method can give various speed and angular command to the navigation module. However, to explore the environment, the subject has to click on a button on the haptic interface to switch to the "Haptic Exploration" mode.

- Scenario 3: Semi-autonomous navigation with multi-modal tele-perception, or "Clickable world". The human subject explores the scene with the haptic interface, and upon finding a target, the subject can click on the position in the haptic space to command the robot to approach that position autonomously. Since the robot navigates on the user's command, this is "semi-autonomous navigation."

The states and events are identically defined in all three scenarios, but only the activation of sub-states of the robotic or haptic modules differ as depicted in Figure 57. The details of the states, events, and substates are explained in Tables 10 through 13.

Table 10: List of States for our Telepresence System.

States	Description
Idle	The initial state. All system modules powered on but only in initial states.
HapticExplore-Static	The human user explores a remote environment with haptic interface while the robot is stationary. The remote environment can be modeled by the Kinect, stereo process, or the mono-stereo process. In the experiments for this Chapter, we limit the source to the Kinect.
HapticExplore-Dynamic	The human user explores the remote environment while the robot is moving. The movements of the robot can be controlled manually by the human user or semi-autonomously by the navigation module based on the goal-position setting by the user.
SSL-Guide	We estimate the location of the sound event through a sound-source localization process and integrate the sound event into spatial feedback through the haptic interface. In simpler terms, when the sound event occurs, the location will be mapped into the 3D map of the haptic workspace and the haptic guiding force will be created toward the position to give the user a perceptual notification that the sound event occurred at that position. The robot is kept static until the human user responds, and has a timeout period of 3 seconds.
Manipulation	The manipulation module is partially activated for the mono-stereo process, and fully activated for object manipulation upon request. For the current state of study, manipulation is not evaluated.

Table 11: List of Events for our Telepresence System.

Events	Description
Start	The system is notified to initiate by the user. For the safety of the experiment setup, a sighted researcher will start the system through the GUI.
End	The system is terminated. It can be due to an emergency stop or completion of a task.
Goal set	A goal is set by the user so the robot can move toward it. A goal can be a position in a 3D map (given by the haptic "click"), or a direction of *left, right, forward, backward* (given by the keyboard commands).
Goal Reached	A robot has reached a goal. It can be determined by the localization module in Scenario 1 or 2, or by a manual keyboard command in Scenario 3.
SSL event	The system has detected a sound event, and the location of the sound-source is estimated.
Ignore	The user has decided to ignore the sound event.
Manipulation start	Manipulation task is requested.
Manipulation end	Manipulation task has finished.

Table 12: List of Sub-states (Robot) for our Telepresence System.

Sub-states	Description
Idle	The robot is stationary, but not powered down.
KeyDrive	The robot is controlled by the arrow keys on the keyboard. "Ctrl" key stops the robot.
HSP(Haptic support planes)-Nav	Another manual drive sub-state with the haptic support plane as illustrated in Figure 56. In addition to controlling the robot to move into four directions, the user can give more refined angular movement command when touching on the four triangular surfaces supported by haptic forces.
AutoNav	The robot autonomously approaches the target position with its localization and navigation modules.
Manipulation	The robotic manipulator is controlled by the haptic interface in the "Manipulation" state, or by the scheduler in the HapticExplore-Static state with "Mono-stereo" Kinect sub-state.

Table 13: List of Sub-states (Haptic source) for our Telepresence System.

Sub-states	Description
Kinect	Kinect is the source for generating the 3D map of the environment.
Stereo	Stereo process generates the 3D map of the environment.
Mono-stereo	Mono-stereo process generates the 3D map of a smaller environment, usually focused on specific objects.

4.3 Experimental Setup

We implemented the system framework discussed in Section 4.1 using our mobile manipulation robot platform (Figure 58), and designed a set of experiments to evaluate the system performance as well as user's experience. The experiments are performed given a common haptic input method for haptic exploration and auditory feedback, but with three distinctive control mechanisms—key control, haptic support plane, and semi-autonomous navigation—to accomplish a goal, which is to "*find a blue/red ball and approach it.*" The spatial setup of a remote scene (5m x 6m) for the experiments is as shown in Figure 59.

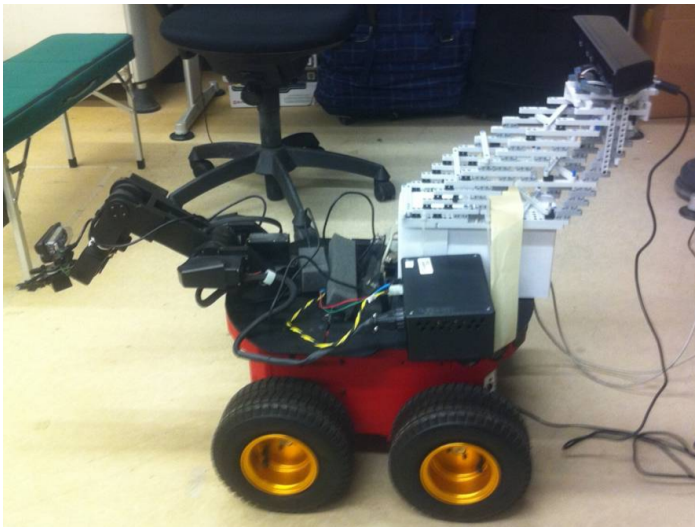
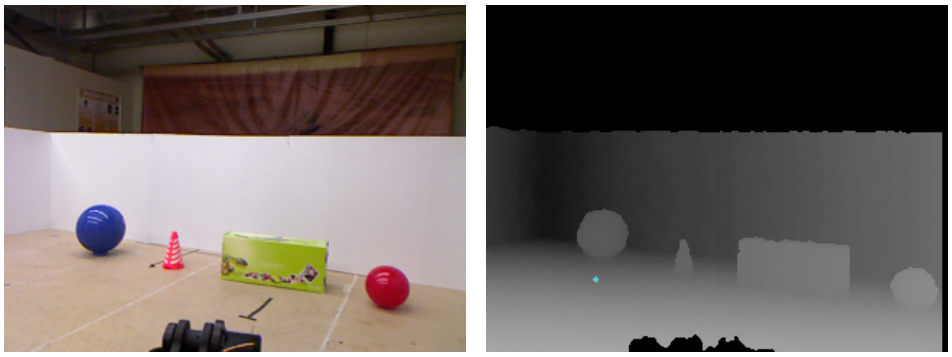


Figure 58: Mobile manipulation robotic system with the Kinect sensor.

The hypothesis we want to study through these experiments are as follows:

- Hypothesis 1: Haptic exploration can benefit telepresence of an individual with a visual impairment.
- Hypothesis 2: Multi-modal feedback for environmental perception can aid in telepresence of an individual with a visual impairment.
- Hypothesis 3: Linkage between control modality and feedback modality can affect the performance of telepresence.



(a) Color view (b) Depth view

Figure 59: Experimental setup.

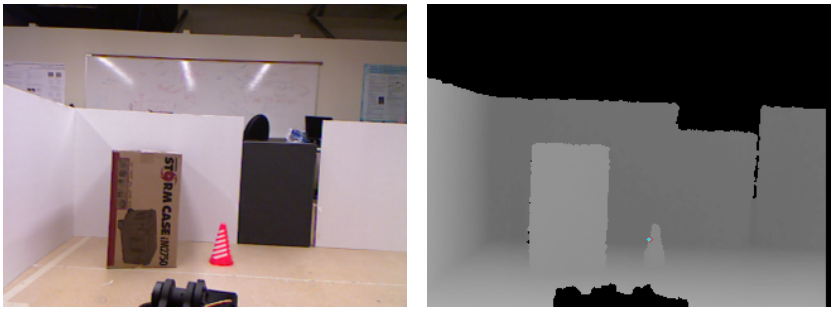
The independent variables we chose are the haptic and auditory feedback method and the control method. The dependent variables are the task completion time, the success rate, and the frequency of giving commands to the robot. As for the controlled variables, we fixed the number of targets and target shapes, which were presented to the human subject.

The procedures for our experiments are as follows:

- Subject consent or verbal assent are acquired prior to the experiment.
- Explanation on the robotic platform, computer system, and the haptic interface is given.
- Tutorial on the haptic exploration: the subject can feel the 3D space as setup in Figure 60.
- Tutorial on the control methods—key control, haptic support plane, and haptic click—are given to the subject.
- Explanation on the auditory feedback—color, distance, and the status of the robot—are provided to the subject.
- The subject starts controlling the robot and accomplishes the task of finding a specific-colored ball (either blue or red ball, whose positions are subject to

switch).

During the experiments, measurements are made on the following criteria: 1) Time for task completion, 2) number of control sequences 3) the trajectory of the haptic probe, and 4) the trajectory of the robot. After the experiments are done, a questionnaire with 10 items are provided for the user to fill out.



(a) Color view (b) Depth view

Figure 60: Tutorial site.

4.4 Results

A total of 10 human subjects (one female and nine male subjects) participated in our experiments. Among them, nine were sighted and one was fully blind. The age group was between 20 and 40. Each subject was given a tutorial of the system (about 5-10 minutes) and then given two tasks (Scenario 1 and 3). Scenario 2 had a issue of mode-switching between control and exploration, thus only users' experience was surveyed after controlling the robot during the tutorial session. The experimental results and analysis are discussed below.

4.4.1 User Performance for Task Completion

To evaluate user performance in task solving with our telepresence system, we measured the control time for task achievement and the success rate for Scenario 1 and 3. The average time taken for Scenario 1 was 58.2 sec, and 32.38 sec for Scenario 3. The median values and the variations are depicted as a bar plot in Figure 61.

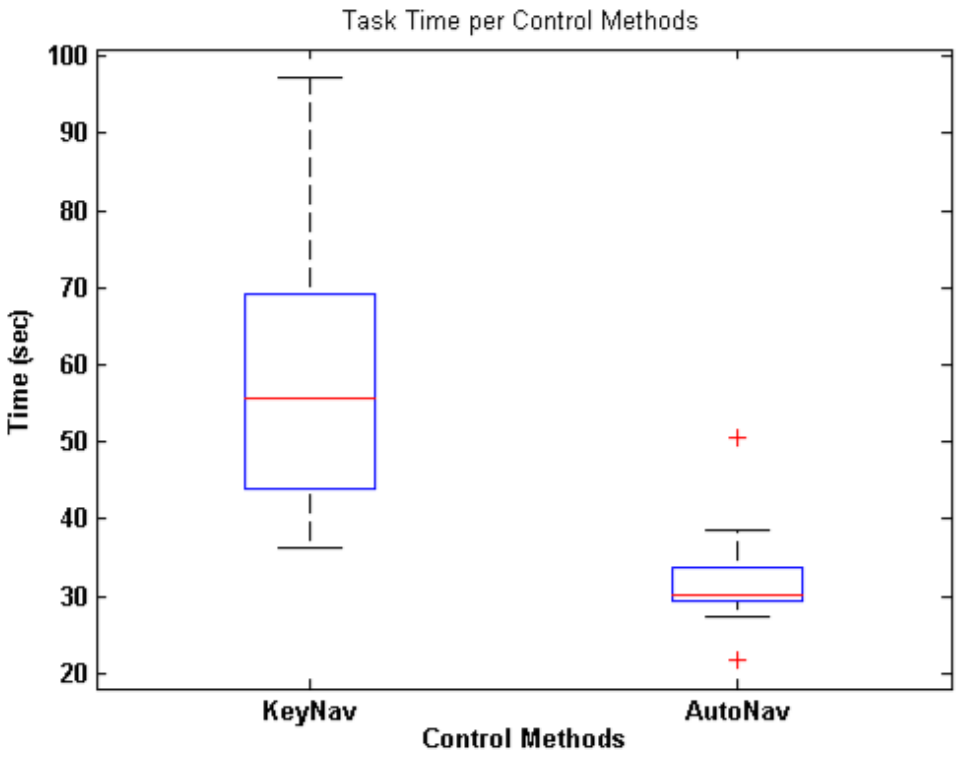


Figure 61: Task time comparison between KeyNav and AutoNav.

The success rates for Scenario 1 and 3 for the 10 participants were 60% and 70% respectively, as shown in Table 14. The average success rate of the participants with visual impairments were 100% for both scenarios, but the number of participants are small at this stage to draw a statistically reasonable conclusion. Based on these results, we can see that the control methods does not affect the performance in the success of the task, but significantly reduce the time for controlling the robot.

Table 14: Success rates of human subjects in Scenario 1 & 3

Subjects Scenario	1	2	3	4	5	6	7	8	9	10
1 (KeyNav)	O	O	O	X	O	O	X	X	O	X
2 (AutoNav)	O	X	X	O	O	X	O	O	O	O

(O: success, X: failure)

4.4.2 Control Load for the User

Another important aspect in human-robot interaction is the factor of control load for the user. We measure this term by counting the frequency of commands on the robot given by the human user, i.e. the number of keyboard strokes in Scenario 1 and the click counts in Scenario 3. The results are shown in Figure 62, which implies that the direct linkage between control space and feedback space plays an important role in the completion of teleoperative tasks.

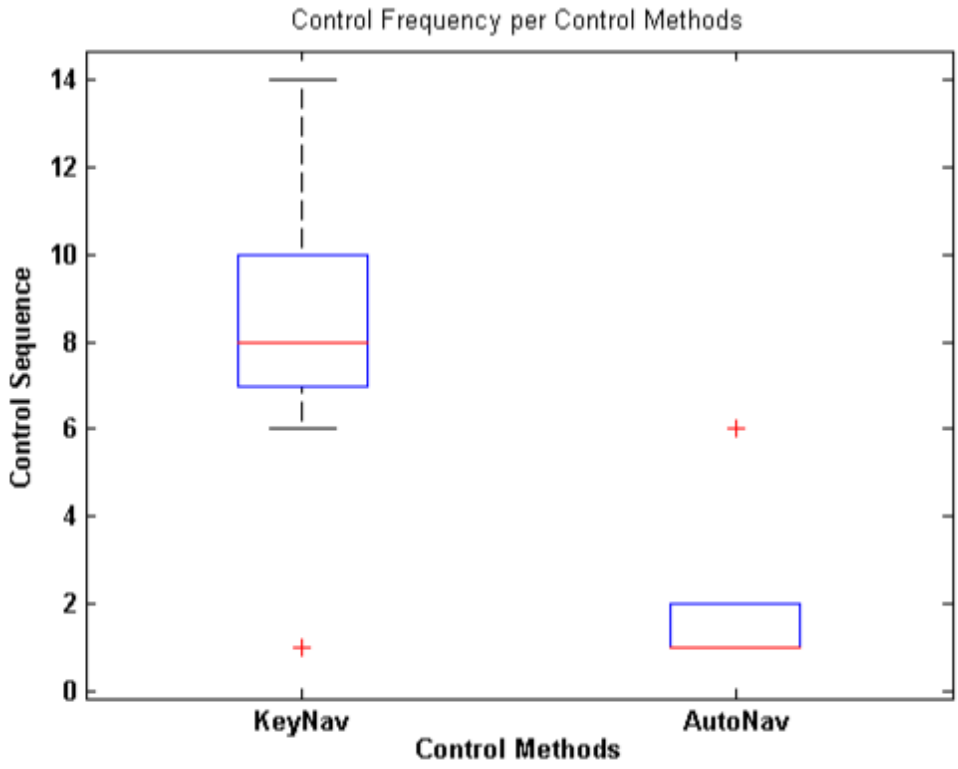


Figure 62: Control frequency comparison between KeyNav and AutoNav.

4.4.3 Robot Trajectory

Exemplary results of user study from Scenario 1 and Scenario 3 is shown in Figure 63 and Figure 64. Sequential images taken from the viewpoint of the robot during the task (at the events of user command) are depicted in Figures 65 and 66. We setup the experimental site in such a way that the robot can perceive all four objects at the

beginning, so there will be no possibility of biasing the scenario. The results show that the semi-autonomous mode achieves the shortest path. Also, by utilizing both haptic and auditory description of the remote scene, the users were able to find the blue ball in both scenarios, meaning that our system achieved tele-perception through multi-modal means.

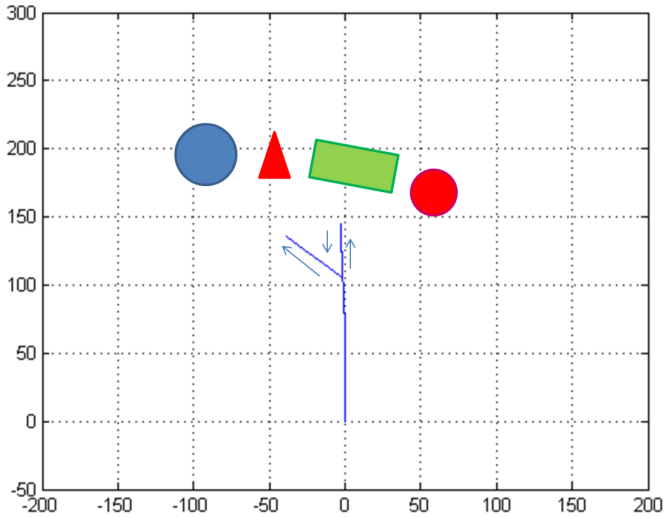


Figure 63: Resulting trajectory of Scenario 1: Haptic Telepresence with Key-control.

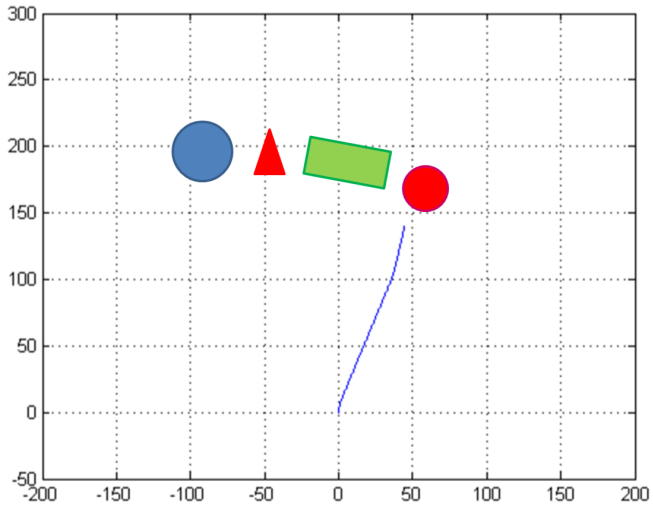


Figure 64: Resulting trajectory of Scenario 3: Haptic Telepresence with Autonomous-navigation.

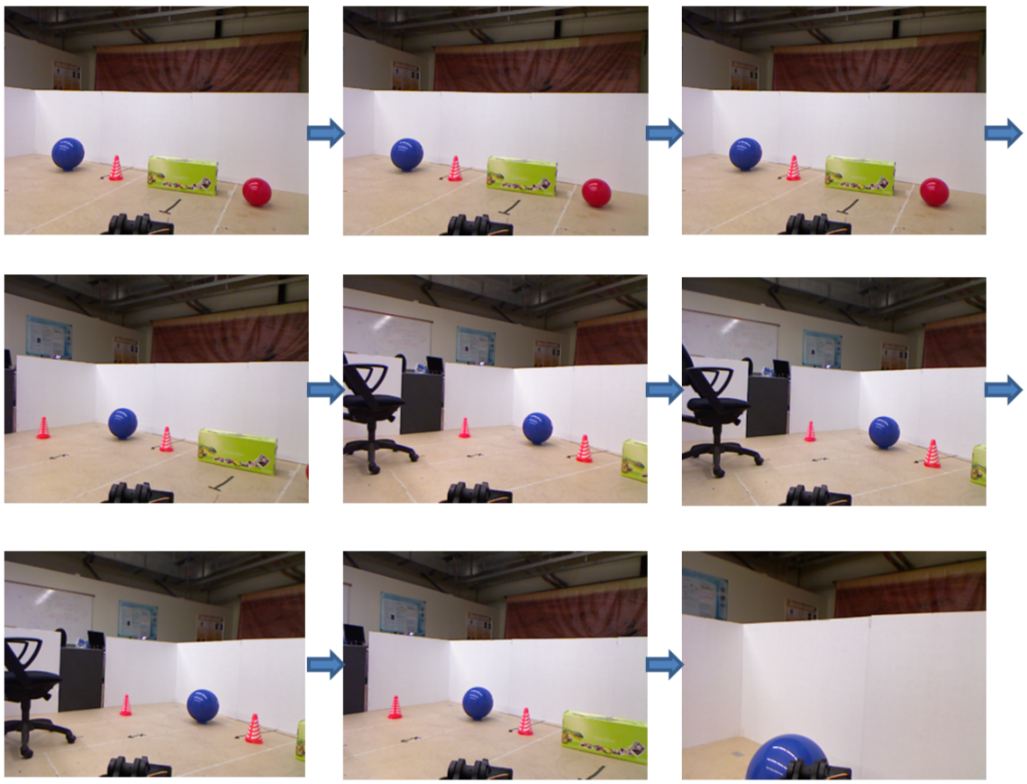


Figure 65: Sequential images on each control commands during Scenario 1: Haptic Telepresence with Key control.

Figure 66: Sequential images on each control commands during Scenario 3: Haptic Telepresence with semi-autonomous path planning.

4.4.4 Questionnaire Results

For qualitative evaluation, participants of our experiments were asked to answer 8 questions. The questions are listed in Table 15.

Table 15: List of color names differentiable with verbal feedback.

Q1: In Exp.1, which Control method was more convenient in controlling the robot? 1. Key control 2. HapticForceSpace 3. Omni-click
Q2: In relation to question #1, which was more intuitive to control (between the first two methods)? 1. Key control 2. HapticForceSpace
Q3: How easy was using the feedback modality in feeling the 3D space? 1. Very Easy 2. Easy 3. Neutral 4. Somewhat difficult 5. Very difficult
Q4: How easy was using the feedback modality in finding objects in the 3D space? 1. Very Easy 2. Easy 3. Neutral 4. Somewhat difficult 5. Very difficult
Q5: In your opinion, what extra feedback information or functionality would be useful for such task as Exp.1? ( )
Q6: How helpful was the verbal feedback with color information? 1. Very helpful 2. Helpful 3. Neutral 4. Somewhat less helpful 5. Not helpful at all
Q7: How helpful was the verbal feedback with distance information? 1. Very helpful 2. Helpful 3. Neutral 4. Somewhat less helpful 5. Not helpful at all
Q8: Did you feel like you are being tele-present in a remote place through the robotic system? How realistic was it, in the degree of 7 to 1? (Very realistic) 7 — 6 — 5 — 4 — 3 — 2 — 1 (Not realistic at all)

Question 1 and 2 were designed to survey the intuitiveness in the control methods. As Figure 67 shows, all participants answered to Question 1 that the semi-autonomous navigation was the most convenient. Also, results from Question 2 show that five person preferred the HSP and three preferred the keyboard control. We will need to recruit more human subjects to draw a statistically significant conclusion. However, the current result implies that haptic modality can provide a more intuitive control method.

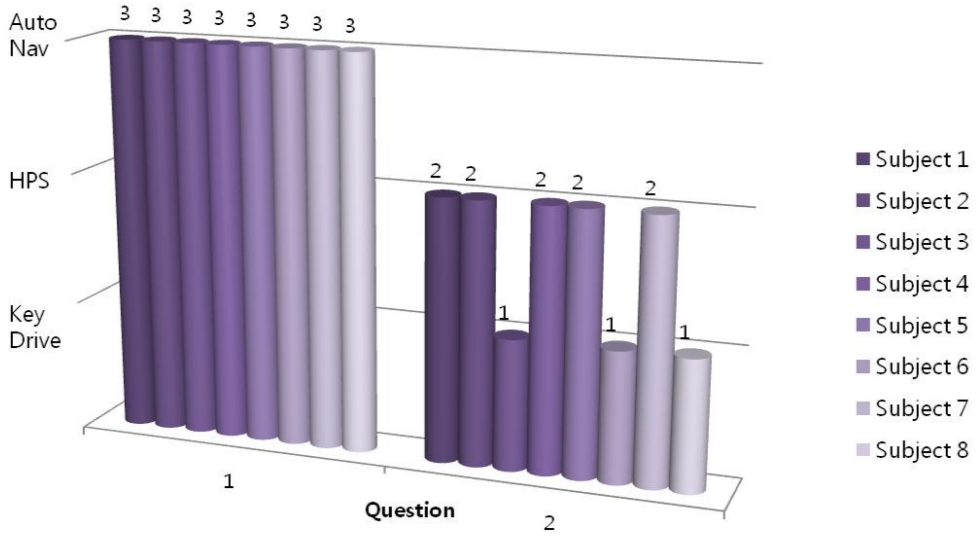


Figure 67: Survey results on question 1 and 2

Questions 3 and 4 asked the participants about the coziness of haptic perception. As the bar-plot in Figure 68 shows, some people felt it was easy to feel with the haptic modality while some others found it somewhat difficult. However, participants found it more easier to explore the remote environment and find objects with haptic feedback.

In addition, Questions 6 and 7 asked the participants' preferences on the verbal feedback with color and distance information, which turned out to be more than helpful as shown in Figure 69.

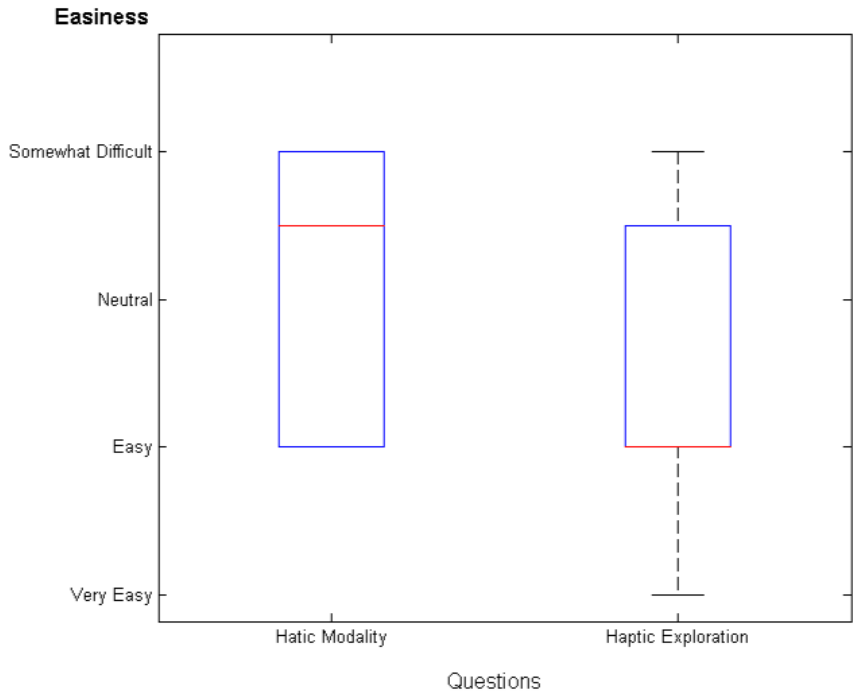


Figure 68: Survey results on question 3 and 4

Finally, we asked the participants' opinion on the level of how realistic the telepresence system was, on the scale of one to seven with one being "not realistic at all" and seven being "very realistic". The answer, as shown in Figure 70, had a distribution between 5 and 6, showing the effectiveness of our system for assisting the visually impaired with remote telepresence and tele-perception.

4.5 Conclusion

We presented our telepresence robotic framework that integrates depth sensing, visual sensing, and haptic rendering process using a FSM to enable real-time tele-perception and tele-operation for individuals with visual impairments. The system was tested with ten human subjects—consisting of both male and female subjects, as well as both sighted and unsighted subjects—under three scenarios. The performance of the users was measured quantitatively with respect to time, success rate, and control frequency. In addition, the user experience was surveyed to qualitatively evaluate

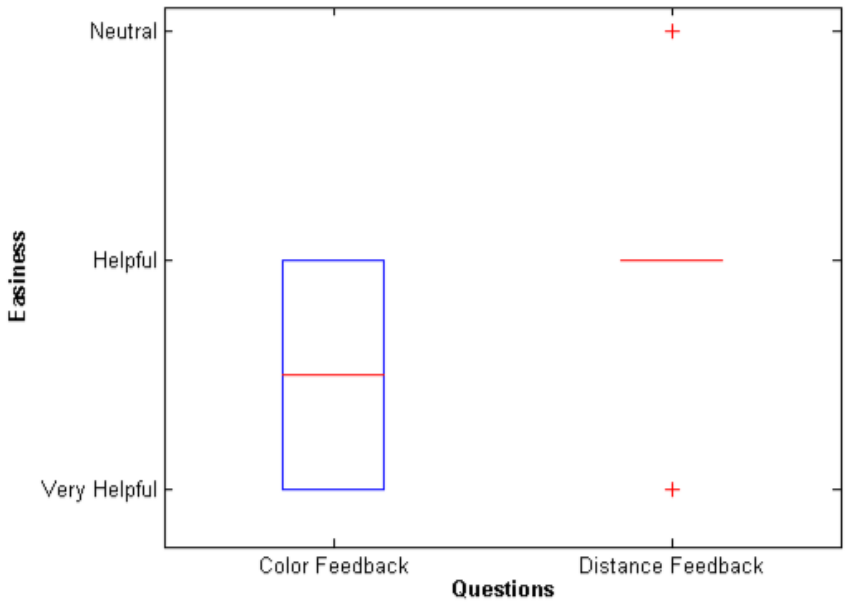


Figure 69: Survey results on question 6 and 7

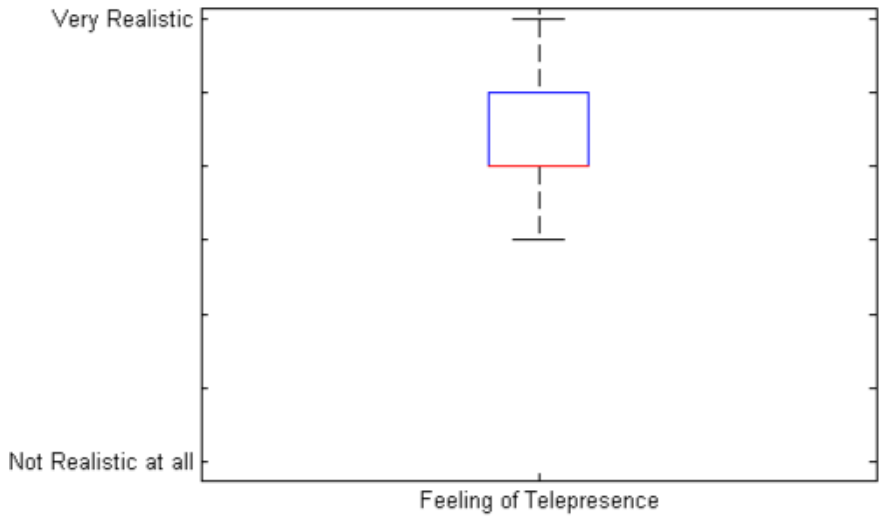


Figure 70: Survey result on question 8

our system. Results show that the combination of haptic exploration of 3D remote environment coupled with semi-autonomous navigation of the robot achieved the highest performance and the most favorable user experience. With this system design, we plan to expand our experimental setup to include remote sound events in the remote environment, and evaluate the possibility of interaction with a remote human being through our telepresence system.

CHAPTER V

HAPTIC SKILL-TRANSFER

Our haptic telepresence robotic system, as presented in Chapters 3 and 4, has one more important aspect for users with visual impairments—the haptic interface itself. While sighted people have numerous ways to learn and to be trained on various skills, individuals with visual impairments have limited resources in learning and training. To address this issue, the haptic interface itself can be used as a training tool for new skills. One good example of this is a study by S. Brewster and A. Crossan [66], in which they utilized a pen-like haptic interface (Phantom^{TM} Omni) to teach visually-impaired students how to draw shapes and how to write letters.

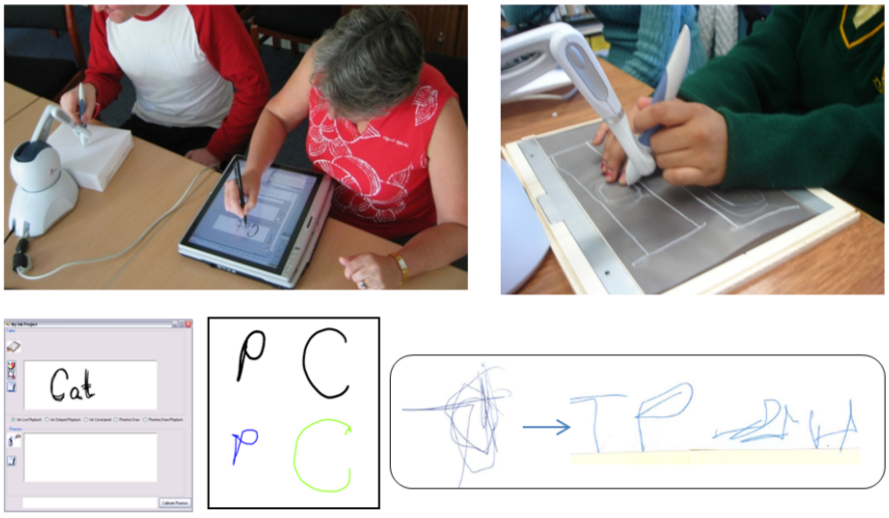


Figure 71: McSig–A multimodal collaborative handwriting trainer for visually-impaired people [66].

However, in this work, the general approach for the transfer of skill is a simple "trajectory feed" of the skill data provided by the teacher. To expand the contents of learning and increase reusability of haptic skill data, we focus on granting learning

capabilities to the system, i.e., the skill data provided by the teacher can be processed with machine learning algorithms to transform into a generalized and compact haptic knowledge. This chapter focus on this issue of a "learning process for haptic skillsets" using our haptic interaction architecture.

5.1 Background

The methodology involved in transferring skills from an expert to non-experts through haptic systems has been well established [54, 55, 89]. A user's skill can be stored, analyzed, and transferred to other human operators by capturing the sophisticated operations of the human expert using a haptic interface with a high degree-of-freedom (DOF). The ability of the haptic device to capture and generate force data makes it both a mediator and a trainer in the skill-transfer process. In the various training architectures used in the skill-transfer process, the expert's data is usually not altered, that is, the reference data is used in the training process without modification. As such, it has led to a common architecture that primarily copies the expert's data rather than compiling the generalized representation of the expert's reference data.

Our approach, on the contrary, focuses on manipulating the expert's data for generalization and transforms the haptic data into a form of "haptic knowledge," as shown in Figure 72. This research is similar to the concept of "learning from demonstration," which is also called "imitation learning" or "programming by demonstration" in the robotics community [62, 76], but is unique in the sense that it uses the haptic pathway to transfer the learned knowledge between the users or between the user and the robot.

The detailed objectives of this research are to integrate the learning path using a haptic interface, increase the efficiency of the haptic-training loop, and expand the modality in human-robot interaction using the haptic pathway. Three primary functional-loops exist in our haptic skill-transfer system: a human-control loop, a

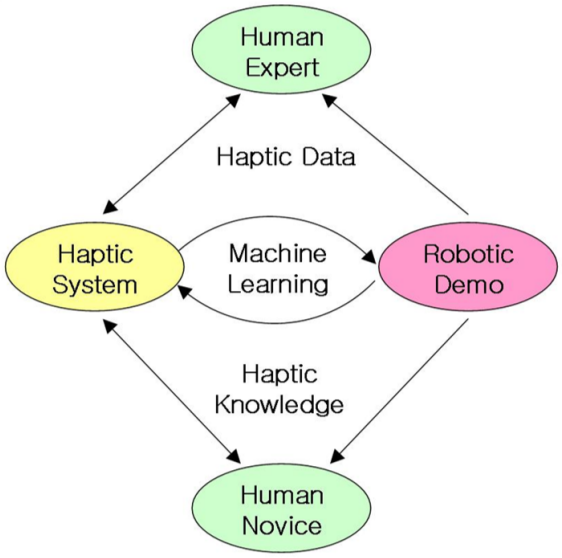


Figure 72: Paradigm of our haptic skill transfer.

robotic-learning loop, and a haptic-training loop. First, the human teacher tele-operates the robot using the haptic interface to perform a task. After the robot follows the commands from the human operator, machine-learning modules are trained over the temporal sequences of the human's operation. Finally, the learned modules are used to control the robotic system and to autonomously perform the trained task. During this process, the haptic device simultaneously generates force-guidance inputs through the haptic device, which are transferred to a novice user, thus completing the haptic-training loop.

5.2 Algorithms

This section explains the machine learning algorithms utilized for processing the skill datasets and a haptic force-feedback methodology to transfer the skill to human users. For the machine-learning framework, two supervised-learning algorithms [23, 33, 69] are employed to process the haptic data provided by a human expert. The learning algorithms are selected based on the ability to generalize over a continuous real-valued

space of input data with high-dimensionality. Our final selections are the multi-layer feed-forward neural network (NN) [26, 48, 52] and the support-vector machine (SVM) [16, 86]. For the haptic force-feedback method, this framework incorporates haptic guidance force-feedback to transfer a time-series 3D skill dataset. Finally, a validation algorithm to analyze the performances of the framework is explained.

5.2.1 Neural Network Learning Module

The first learning algorithm we implement is a neural-network regression algorithm, which enables learning in a 3D Cartesian space. Time-series data patterns are collected from the human user's control sequences, and used recursively to train the multi-layer feed-forward NN modules. As Figure 73 illustrates, the inputs to the NN modules are as follows: (1) the current end position of the robotic arm in 3D coordinates, (2) the differential values of the current and previous positions (representing the velocity of the end position of the robotic arm), and (3) the time-steps from the beginning of the sequence. The time-step input is needed to differentiate between states when there are intersections or circular motions in the trajectory of the robot.

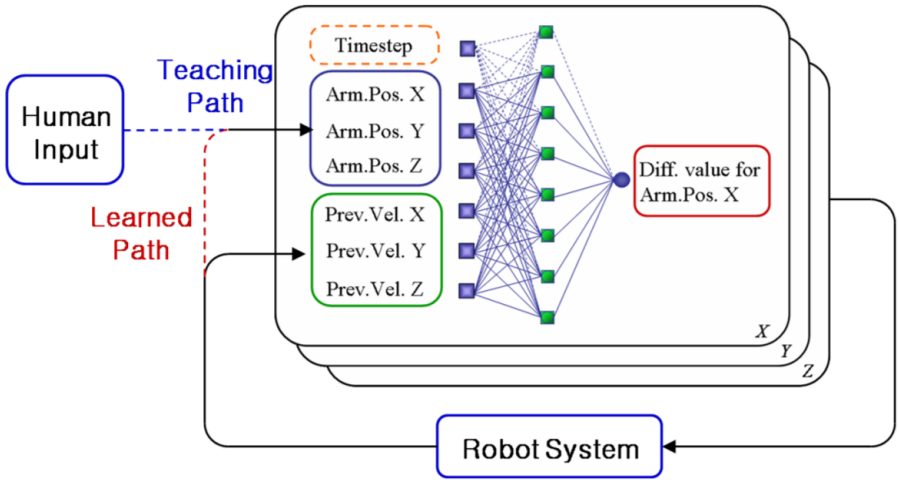


Figure 73: Structure of the learning module with multilayer neural network.

For validation, we train and test the NN module to learn trajectories associated with the expert's input to the haptic device. The back-propagation method [65, 71]

is employed to optimize the learning process, and the training is terminated if the error term reaches a certain allowable criteria. The networks then go through both a preliminary evaluation process and a real-system evaluation process. The details of these processes are explained in Section 3.3.

5.2.2 SVM Learning Module

SVM [16,86] is another popular machine learning algorithm for data classification and dimensionality-reduction process. SVM uses a kernel method to find a hyper-plane between high-dimensional training datasets. SVM is relatively fast compared with other classification algorithms, and also shows strong resilience against disturbances while maintaining short time for training.

The epsilon support-vector-regression (ϵ-SVR) approach is used to map seven-dimensional real-valued inputs to one-dimensional real-valued outputs. This forms a set of SVM classifications over the given input and output spaces. The objectives of ϵ-SVR in a given training set of N instance pairs (x_i, y_i), where $i = 1, \cdots, N$ and $x_i \in R^N$, are as follows: (1) to find a function $f(x)$ that has at most ϵ deviation from the obtained actual target y_i for all the training data, and (2) to make the hyper-plane as flat as possible. Specifically, the following optimization problem must be solved in ϵ-SVR: the support vector w, as defined in Equation 31, is subject to the condition defined in Equation 32.

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^{N} \zeta_i \tag{31}$$

$$y_i \left(w^T \varphi \left(x_i \right) + b \right) \geq 1 - \zeta_i, \ \zeta_i \geq 0 \tag{32}$$

The term w is the weight coefficient vector of ϵ-SVR, ξ_i is the slack variable, and $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$ is the kernel function that maps the training vector x_i to

a high-dimensional space using the transformation function $\Phi(x_i)$. SVM is used to find a separating hyperplane with a maximum margin in this high-dimensional space. The kernel functions used are as follows:

$$\text{Linear kernel}: \ K(x_i, x_j) = x_i^T x_j, \tag{33}$$

$$\text{RBF kernel}: \ K(x_i, x_j) = e^{-\gamma |x_i - x_j|^2}, \ \gamma > 0. \tag{34}$$

The parameters C and γ significantly affect the accuracy of the SVM modules, where C is the positive-valued penalty parameter of the error term and γ is the kernel parameter. The grid-search method is adopted to find the optimal values for C and γ in a reasonable amount of time [13].

5.2.3 Haptic Guidance

Forces for haptic guidance are generated both in the human control cycle and in the haptic training cycle. During the human control cycle, the haptic device creates a passive guidance force feedback which generates potential-like centering forces to the operator's hand position. The primary objective of this guidance is to provide a reliable control environment, since it is difficult for a human operator to maintain a steady position in a 3 dimensional workspace without any support in the device. The haptic device is updated every 10 ms (100 Hz), so as the human operator moves the haptic device, the passive potential force follows the human operator's position and creates continuous holding forces, enabling a passive haptic support.

In the haptic training loop, after the learning cycle, a guiding force field is also applied to the device, except that this time the learning module generates a potential-like force field based on the current position and velocity. In this regard, the haptic force guidance becomes an active guidance toward the next position, thus guiding the human trainee to follow the same trajectory as the human expert.

5.3 Experimental Setup

We setup our haptic telepresence system for human operators so the human "expert" can hold the haptic interface to provide a training sample trajectory, which can be demonstrated by the robotic system as well. The expert's data can be stored and processed by the machine learning algorithms explained above, and after learning, the learned haptic knowledge will be reproduced by both the haptic interface and the robotic system.

To validate our approach, human handwriting is selected as the test scenario of our experiments, because it functions as a good platform to showcase the complexity of human dexterity. Handwriting is a research topic studied in haptics for training applications [87], and it is also a challenging area in robotics [67, 68]. Handwriting is difficult to study because it is a three-dimensional (3D) task, which increases to six dimensions if changes in the orientation of the pen are considered. Most importantly, the position and velocity of the pen can change abruptly over the time domain, which makes pattern generalization even more complex.

The data to be learned is constructed as a set of arrays of real valued data taken over time. The human control data consists of a set of trajectories over 3D space, and the related data that we learn are the set of motion vectors corresponding to the trajectories from current positions to the next positions. Computed over the time-series data over N timesteps, the final learned result becomes a function F as described in Eq. 38.

$$P_i^C = \left(p_x^C, p_y^C, p_x^C\right) : \text{current position at time } i, \tag{35}$$

$$V_i^P = \left(p_x^P, p_x^P, p_z^P\right) : \text{previous motion vector at time } i, \tag{36}$$

$$V_i^N = \left(p_x^N, p_y^N, p_z^N\right) : \text{next motion vector to be taken at time } i+1, \tag{37}$$

$$F(P_i^C, V_i^P) = V_i^N : \text{learned mapping with NN / SVM for time } i. \tag{38}$$

To test our algorithms, we select letters "3", "b", and a word "ML". The letters

"3" and "b" are chosen since the "3" is written in one continuous stroke consisting of two similar curves, and "b" is selected since it contains two basic strokes, a straight line and a circle with a sharp turn in between, sharing an intersection. These two letters are thus suitable for evaluating whether the NN and the SVM learning modules are capable of learning spatiotemporal patterns. The word "ML", abbreviation for 'Machine Learning', is selected to validate if the learning modules can learn to write multiple letters as a whole sequence.'

For the SVM training and implementation process, we utilize the LibSVM [13] to implement the SVM learner in our system. LibSVM provides basic SVM functions in an integrated software package for support vector classification, regression, and distribution estimation.

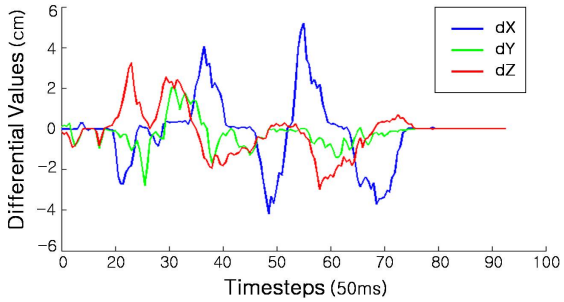
5.4 Results

After setting up the system, a preliminary experiment was initiated to see if the machine-learning system was able to learn a trajectory pattern. The performances of the NN and SVM modules are investigated during this experiment. A training example of writing the number "3" was given to the learning modules as shown in Figure 74.

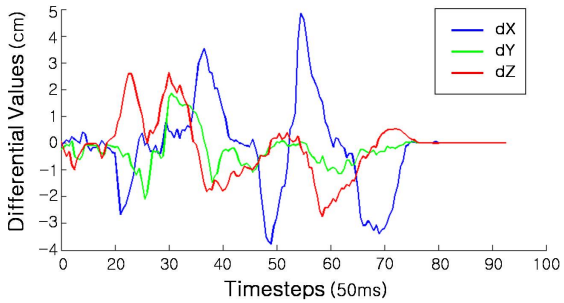
Figure 74: Robotic arm writing "3" after learning.

After training, outputs from the learning modules were observed using the same

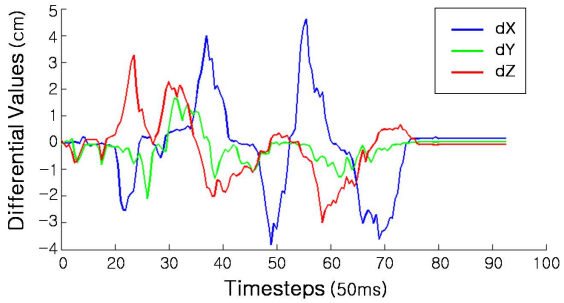
input data. This method was not a cross-validation process. However, the main goal of this preliminary experiment was to check if our learning architecture could actually perform regression over complex 3D patterns with only a single training data sample. The results were quite satisfactory, and the NN module showed an output pattern generalized from the initial data. The SVM module also generated the pattern successfully, and even more details in the pattern were learned as shown in Figure 75.



(a) Human's writing trajectory in X,Y,Z-domain.



(b) Neural-network module controlling the robotic arm to write the learned letter.

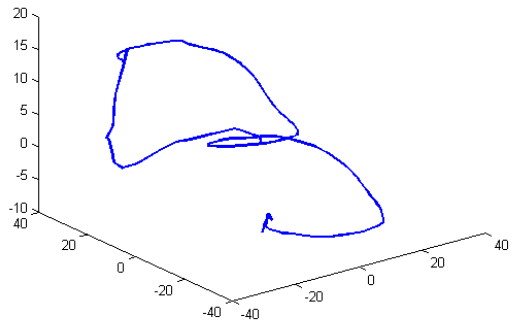


(c) SVM module controlling the robotic arm to write the learned letter.

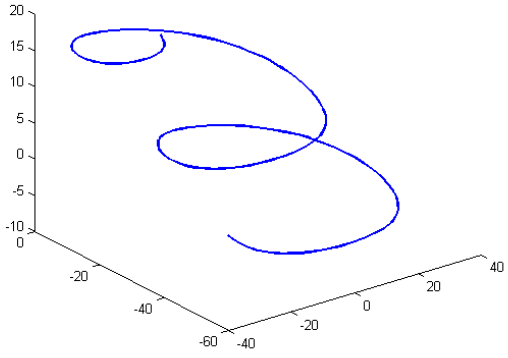
Figure 75: Preliminary results for learning of writing "3".

Our NN modules are constructed with seven input nodes, a selected number of

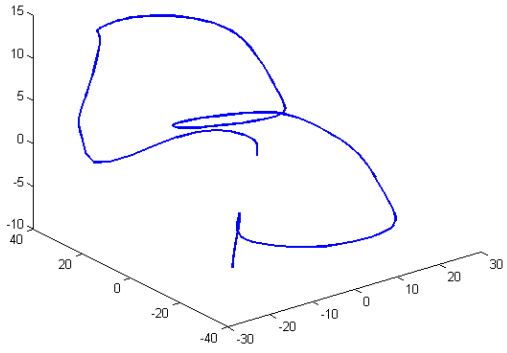
hidden-layer nodes (from the set of 4,8,12,16), and one output node for each X, Y, and Z coordinate. The other parameters include a learning rate of 0.15, a momentum term of 0.85, an error limit of 5%, and an iteration count of 3000, with minor variances depending on the dataset. Figure 76 shows that the NN module learned the pattern of writing "3" (Figure 76b) with a single input pattern (Figure 76a). The trajectory promptly regenerated the two sequential curves for the number "3", similar to how a human would write "3" although the trajectory was smoother than the human's pattern.



(a) Human expertś handwriting (training data)



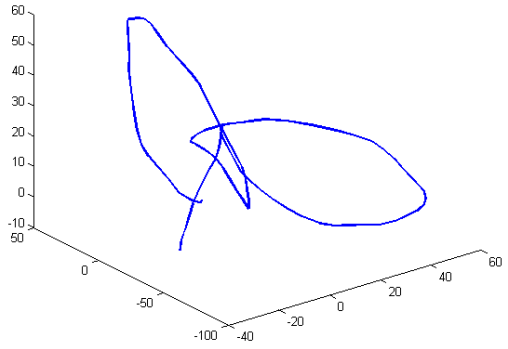
(b) NN module result



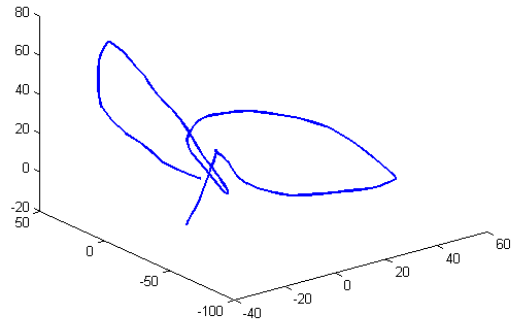
(c) SVM module result

Figure 76: Learning results for "3".

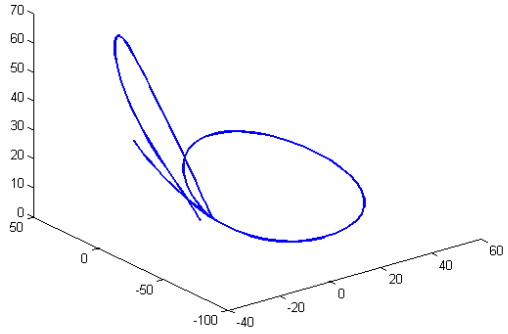
The letter "b" was then demonstrated twice by the human expert using the haptic



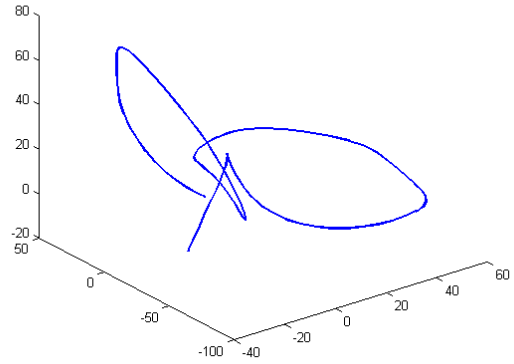
(a) Training data 1



(b) Training data 2



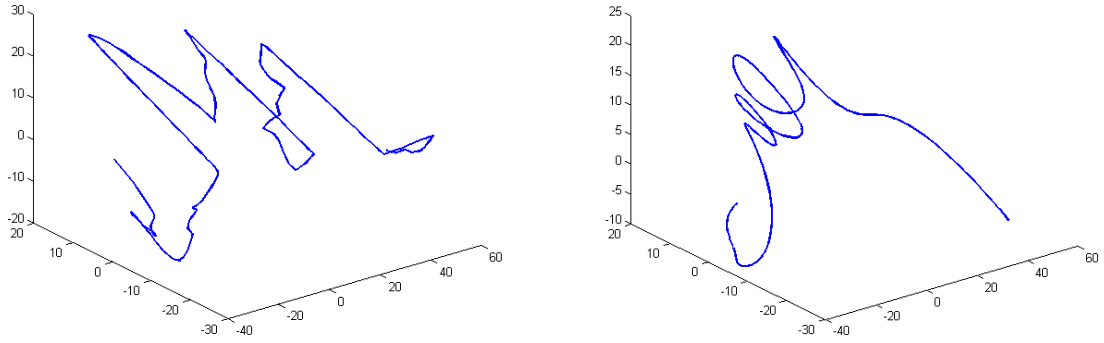
(c) NN module result



(d) SVM module result

Figure 77: Learning results for "b".

interface (Figure 77a and 77b), and the movement of the robot was recorded and fed into the NN modules with and without the time-step inputs. When the learning process was completed without the time-step information, the robot miscalculated the first straight downward-stroke and drew a slightly bent curve. After finishing the circle, the robot kept repeating the circle stroke, because the velocity and orientation inputs matched whenever it returned to the same starting point. However, the robot drew a perfect straight line at the first stage and reduced its speed after drawing one circle stroke when the time parameter was used as an input vector to the NN module



(a) Training data (b) NN module result

Figure 78: Learning results for “ML”.

(Figure 77c). Lastly, a character set “ML” was provided to the NN learner, and after a few minutes of training, the robot started to write a deformed “M” followed by a smoothed out “L” (Figure 78).

The only parameter that was changed for the NN modules in learning of the three letters was the number of middle-layer nodes. In our framework, the middle layer was basically composed of four nodes. The four middle nodes worked well with connected characters such as “b” and “3”. When the nonlinearity in the pattern increased with more complex letters such as “M” or “A”, we increased the number of hidden layer nodes to successfully train the NN module.

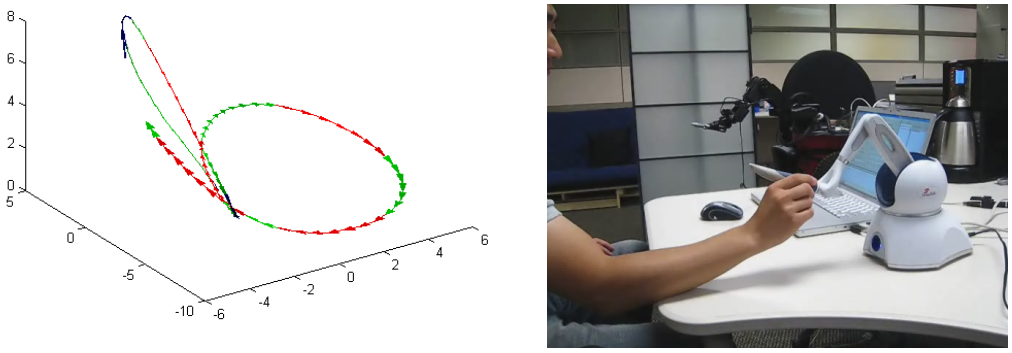
The results from the SVM process are also shown in Figure 76c and Figure 77d, which show that the SVM modules successfully learned to draw more elaborate patterns compared to the NN module, recreating the details of human handwriting. When the robot wrote the letter “b”, the SVM learner stopped moving exactly when it reached the starting point of the circle stroke. When writing “3”, the SVM module recreated the personal style in the handwriting of the human expert. However, for the complex pattern “ML”, the SVM learner was not able to accurately write the letter. The grid search could not find a proper combination for the parameters, and

the SVM module could not write any letter.

In the grid search for the SVM learner, a set of (C, γ) pairs is tested over a predetermined range, and the one with the best cross-validation accuracy is selected. Specifically, the pair that generates the least mean-squared-error (MSE) is kept per iteration. Exponentially increasing the sequence of C and γ values is a well-known method for identifying good parameters. The optimal C and γ pair is searched exhaustively over the range of 2-10 to 210. One of the advantages of this approach is that the search can easily be parallelized, because each (C, γ) pair is independent. Moreover, scaling can also be used to derive better outputs. The dataset is scaled down before training and scaled up when the learning module is in control. Two positive results were achieved using the scaling method: The method created more precise training results, and it also shortened the training time dramatically. Consequently, the training time decreased from minutes to seconds when a linear kernel was used for training.

Finally, a sequence of force vectors for haptic guidance (Figure 79a) was generated in both the human-control and haptic-training cycles. During the human-control cycle, the haptic device created a passive guidance with force feedback that generated potential-like centering forces to the operator's hand position. The primary objective of this guidance was to provide reliability in the control environment, because maintaining a steady position in a 3D workspace without any support in the device is difficult for a human operator. Therefore, as the human operator moved the haptic device, the passive potential force followed the human operator's position and created continuous holding forces. In the haptic-training loop, a guiding-force field was applied to the device after the learning cycle, although this time the learning module generated a potential-like force field based on the current position and velocity. In this regard, the haptic force-guidance becomes an active guidance toward the next position. Through these processes, the human trainee is guided to follow the same

trajectory as the human expert, learning how to write a letter like the expert teacher.



(a) Haptic trajectory (b) Synchronized robotic-motion

Figure 79: Haptic-force-guidance trajectory and robotic-motion synchronized haptic skill transfer.

5.5 Analysis of Haptic Skill Transfer

The first factor considered for the evaluation of our skill-transfer system is the ability to compress the data, which shows the effectiveness of applying machine learning algorithms to a robotic problem. Figure 80 and Table 16 show that the original pattern data in writing "b", "3", and "ML" were 27, 22, and 36 kilo-bytes (KB), respectively. However, the trained NN modules downsized them to two, four, and six kilo-bytes, respectively, whereas the SVMs only reduced the size by only about 70%, resulting in 17, 14, and 21 KB, respectively.

Table 16: Data compression ratio of ANN and SVM over 3 datasets (UNIT=KB).

Learner	"3"	"b"	"ML"
NN	2	4	6
SVM	17	14	21
Original Data	27	22	36

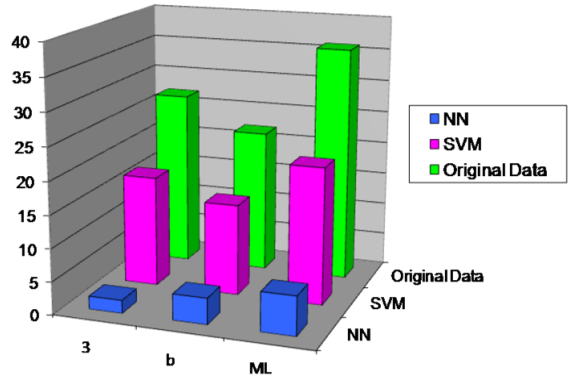


Figure 80: Data compression results of learning modules.

The second feature for observation was the training time. The training time for the SVM regression was surprisingly fast compared with the NN learner, which is well-known for its long training time (Figure 81 and Table 17). The NN learner usually took 10 to 30 times longer to train than the SVM modules, whereas the SVM learner usually finished the training in a few seconds. However, SVM learners are very sensitive to the parameters (especially with C and γ) on each dataset, whereas the NN learners can be trained over several datasets using the same parameters. Thus, the grid-search was performed for every collected dataset, and the total time taken for the parameter search was around 1 min.

Table 17: Training time for ANN and SVM over 3 datasets (UNIT=sec).

Learner	"3"	"b"	"ML"
NN	57	62	183
SVM	1.39	5.62	9.31

The last parameter observed was the measure of similarity, that is, how well the learning module learned from what the human teacher taught (and with only a few teaching trials). We adopt and implement a LCS (Longest Common Subsequence) algorithm [4] to compare the two sequences with possibly different lengths, which was

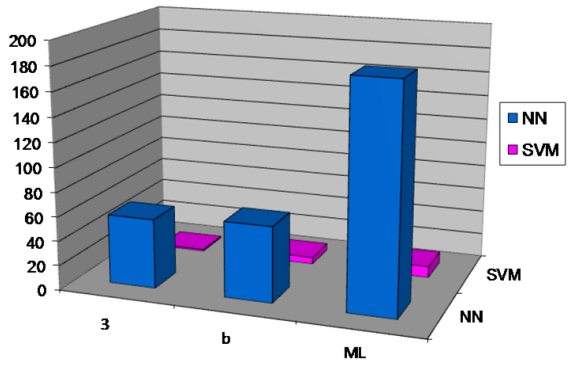


Figure 81: Training time results of learning modules.

originally designed to compare common phrases in texts.

We modified the algorithm to measure the differences in two vectors by extracting the longest sequence of similar trajectory slices. This algorithm increases the number of 'hits'if the error between the two trajectory slices is within a certain threshold, and keeps the maximal value of the 'hits' if not. With this method, we measure the similarity between the original pattern and the pattern created by the robot, and provide a computational match between the two writings.

The algorithm is described in detail in Eq. 39. For the comparing of a sequence $A(i)$ and an original sequence $B(i)$ given a certain geometric distance threshold δ for positions in 3D cartesian space,

$$LCS(A,B) = \begin{cases} 0, & \text{if } A(i), B(i) = null \\ 1 + LCS(A(i-1), B(i-1)), & \text{if } |A(i) - B(i)| < \delta \\ \max(LCS(A(i), B(i-1)), LCS(A(i-1), B(i))), & \text{otherwise.} \end{cases} \tag{39}$$

Table 18 shows that the LCS results for the character "3" were 65.7% and 87.7% for NN and SVM learners, respectively. The result of the SVM module was closer to the features of the original data, although both learners legibly wrote the number "3". The results from the similarity measure algorithm also support the current experimental results. For "b", both algorithms created good results. The LCS result

Table 18: LCS results for ANN and SVM.

Learner	"3"	"b"	"ML"
NN	65.7%	83.5%	42.9%
SVM	87.7%	85.6%	–

of the SVM module was higher than the ones from NN. This proves that the SVM modules learned more details than NN modules, capturing the personal style in the handwriting of the human expert. However, both algorithms also have limitations in overly complex input data. Only a 42.9% match was achieved from the NN, and the SVM learner failed to execute the task at all, when challenged to learn the letters "ML". Nevertheless, this does not imply the learning modules are incapable of learning of human handwriting, but suggests that the learning of handwriting is a character-based task.

5.6 Conclusion

This chapter has introduced an application of our telepresence system as a training station of new skills to the user through the haptic interface. The expert's training trajectory for a new skill is processed by the machine learning algorithms to transform the haptic skill data into compact haptic knowledge. Results show that our system successfully creates generalized haptic knowledge from only one or two training examples, and the system is capable of transferring the haptic knowledge both through the haptic force guidance as well as through the robotic demonstration. As expected, the SVM results in learning more detailed and accurate strokes than the NN, since the SVM algorithm tries to find more optimal classifiers by keeping specific features. However, the NN exhibits better results if the sequence gets more complex, due to the NNs powerful capabilities in generalization.

CHAPTER VI

CONCLUSION

For individuals with visual impairments, the ability to navigate in real-world environments becomes a challenge, and the ability to perceive objects within the environment involves the use of non-visual cues, which typically requires direct contact with the world. This research focuses on the problem of utilizing haptic modality to create an interactive channel for communication between a robotic system and the user to aid in ahcieving tele-perception and tele-operability in a remote environment. To be more specific, this research provides a novel framework for a visually-impaired user to expand the limitations of one's physical living area through a telepresence robotic assistant using a haptic interface. The major contributions to this effort include: 1) study and design of a haptic modality as a non-visual feedback method for in-situ perception transfer, 2) design and evaluation of a vision-based haptic exploration framework that generates 3D haptic perception for representing real-world 3D spatial information, and 3) design and evaluation of a real-time interactive telepresence system for the visually impaired using an active depth camera, a color vision sensor, audio sensors, a mobile manipulator robotic system, and a haptic interface.

Additionally, the aspects of learning and training for visually-impaired users through the haptic interface have been studied. Differentiated from the traditional approach of "trajectory feeding", our system is capable of processing the teacher's haptic trajectories with machine learning algorithms to learn the general patterns. Using this more compact "haptic kwowledge," our system can regenerate the teacher's trajectory more efficiently to guide the students with the haptic interface to teach new skills.

6.1 Key Contributions

6.1.1 Haptics for Non-visual Environmental Feedback

The first contribution of this research is the evaluation of the effectiveness of haptic feedback as a primary means for non-visual feedback for the visually impaired. We designed a telerobotic system and a non-visual framework for transferring the robot's in-situ perception. The non-visual feedback signals enabled a person with a visual impairment to access a robot in a remote location and get in-situ information from the robot. The system design included a human-robot interaction framework that utilizes haptic and auditory feedback to transfer a robot's motion and sensor status to a human user.

We setup an experiment that measured the accuracy of human perception in sensing the status of a remote robot using haptic and auditory feedback, while the subject was being taught to learn how to program a robot. Participants were grouped into teams after training sessions, and were given tasks to program their robot to run through a maze based on the initial experience of haptic and auditory feedback from a solution robot running through the maze. The dynamic time warping (DTW) algorithm [74] was incorporated to measure the match between the students' output and the reference. The learning curve of the human subjects in a given task was also observed. Results show that the use of the multi-modal feedback, such as haptic and auditory feedback, provided better perceptual feedback and enabled more effective learning. After determining their perceptual capabilities, we measured their learning curves as the trial runs increase, and found no statistical difference in learning between sighted users and unsighted users with our system. This suggests that our multi-modal feedback system is serving its design objectives—to enable a visually-impaired person to gain knowledge of a remote environment through a robotic system.

6.1.2 Haptic Exploration from Robotic Vision

Given the effectiveness of haptic feedback, the second contribution of this work is developing algorithms needed to enable 3D haptic exploration directly from robotic visual perception. In order to achieve this objective, a haptic linkage is created between an assistive robotic system and a human user by delivering haptic perceptions of a remote environment, based on the sensory data acquired by a mobile robotic platform. To accomplish this, the user is provided with the ability to haptically explore the real-world environment at multiple scales through the "eyes" of a robotic system.

The challenges that were addressed by this system involved 1) recognizing the environment in multi-scale dimensions since real-world objects are of various sizes and located at different three-dimensional world positions, 2) developing a system architecture for fast processing of the vision sensor data, and 3) enabling three-dimensional haptic interaction for smooth tactile representation of objects in the environment.

A haptic exploratory mobile manipulation (HEMM) system was constructed with a stereo camera, a monocular camera, a mobile manipulator robot, and a haptic interface. The software framework for governing the visual sensing process and robot control has been presented, and the performance has been measured. Three objects were modeled instantly with our system, and the haptic rendering of those model was experienced by human subjects, followed by questions to identify the objects in a given set of items.

Subjects with no visual impairments showed 79% to 84% recognition rate for the three objects, while subjects with visual impairment rated 87.5% for each objects. We were able to generate a 3D haptic model using robotic visual sensors with a small processing time (in the order of seconds) and represent this model to the human user in real-time using haptic perception. Human subjects were able to feel a remote object using the HEMM system and were also able to distinguish objects using only

haptic exploration.

6.1.3 Haptic Telepresence and Multi-modal Interaction

Given direct 3D haptic exploration from robotic visual perception, our third contribution was addressing the challenging issue of developing a telepresence robotic system for individuals with visual impairments. For this purpose, the Kinect sensor was utilized to provide active depth sensing. Using the same haptic device, this telepresence robot can simultaneously model a remote environment in a 3D haptic space as well as be controlled by the user. Through this novel system design, the goal of this research is to extend the boundaries for activities of daily living for individuals with visual impairments.

Our telepresence robotic framework integrated depth sensing, visual sensing, and haptic rendering process to enable real-time tele-perception and tele-operation for individuals with visual impairments. The system was tested with 10 human subjects under three scenarios. The performance of the users was measured quantitatively with respect to time, success rate, and control frequency. In addition, the user experience was surveyed with questionnaire to qualitatively evaluate our system. Results show that the combination of haptic exploration of 3D remote environment coupled with semi-autonomous navigation of the robot achieved the highest performance and the most favorable user experience. Results showed that users found our system fairly helpful in sensing a remote environment and finding objects whether they felt comfortable or uncomfortable with haptic modality, and found the verbal feedback of color and distance to be very helpful. Based on the task success rate and time measurements, we could conclude that autonomous path-planning from the robotic system is greatly needed, as well as the direct mapping between exploration and control. Additionally, the user's experience with our telepresence system turned out to convey fairly realistic perception of telepresence to the user.

6.1.4 Haptic Skill-Transfer

Our haptic telepresence robotic system provides one final important aspect for users with visual impairments, that is, using the haptic interface as a training station for learning new skills. Instead of conventional method of "trajectory feed" that used the original expert's training trajectory for a skill, we incorporated machine learning algorithms to process the expert's data and transform it into a generalized and compact haptic knowledge.

Results showed that our haptic skill learning architecture successfully downsized the memory for storing the skill data—about 80% reduction with the artificial neural-network (ANN) algorithm and 30% with the support-vector machine (SVM)—and achieved high learning performances—about 65% in the case of ANN and 88% with SVM. In terms of training time, ANN took about 20-30 times longer time than SVM to be trained over the given data. This haptic skill-transfer architecture transformed the haptic data into compact haptic knowledge and successfully createed generalized haptic skill patterns from only one or two training examples.

6.2 Future Work

This research was performed in collaboration with the National Foundation for the Blind (NFB), the Center for the Visually Impaired (CVI) in Atlanta, GA, and Wiz-Kidz Technology Center. Through this collaboration, we were able to work with participants with varying degrees of visual impairments, who gave us a lot of feedback on our system. The next logical step in the progression of this research is to address that feedback and provide an upgraded system with better assistive functionalities. The first task will be to design and incorporate a larger robotic platform—a more dexterous robotic manipulator with a larger mobile base—to work in a human-centered environment. The current robotic platform is designed to test and evaluate algorithmic design and issues, not for performing more complex assistive tasks.

Furthermore, there are optimization issues that need to be resolved with the control and feedback channels. The current channel for teleoperation is based on a wired-network system to cope with the large bandwidth from the vision sensors. By adapting ideas from the teleoperation research fields such as in Ryu et. al. [27, 73] or computer vision area for efficient point-cloud data management [24], our system may function efficiently over a wireless network.

Finally, the generation of global perception and its transference to the remote user can greatly improve the realistic experience of our telepresence system. The haptic exploration presented in this thesis is based on the local perception from the view-point of a remote robot system, which causes extra cognitive load for the human operator when the robot is moving. Visual-SLAM based global map building techniques will enable us in this task, and more thoughtful design of auditory or verbal feedback for the notification of the robot's global position and orientation will make the interaction more robust. A more futuristic approach is to develop a learning agent that monitors human's interaction in the remote space and predicts human intentions. Combined with semantic knowledge of the environment, our system will be able to provide more intelligent and sophisticated assistance to individuals with impairments.

REFERENCES

[1] K-Sonar. Bay Advanced Technologies Ltd., http://www.batforblind.co.nz/how-ksonar-works.php.

[2] Miniguide. GDP Research, http://www.gdp-research.com.au/minig_1.htm.

[3] Wearable Haptics for Virtual Reality. Robotics and Biology Laboratory, Technische Universtät Berlin, http://www.robotics.tu-berlin.de/menue/research/wearable_haptics_for_virtual_reality/.

[4] A. Apostolico and C. Guerra. The longest common subsequence problem revisited. *Algorithmica*, 2(1):315–336, 1987.

[5] R.S. Avila and L.M. Sobierajski. A haptic interaction method for volume visualization. In *Visualization'96. Proceedings.*, pages 197–204. IEEE, 2009.

[6] G.H. Ballantyne and F. Moll. The da Vinci telerobotic surgical system: the virtual operative field and telepresence surgery. *The Surgical clinics of North America*, 83(6):1293, 2003.

[7] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2002.

[9] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[10] J. Broeren, M. Rydmark, and K. Stibrant Sunnerhagen. Virtual reality and haptics as a training device for movement rehabilitation after stroke: a single-case study. *Archives of physical medicine and rehabilitation*, 85(8):1247–1250, 2004.

[11] G. Buja and G. Indri. Improvement of pulse width modulation techniques. *Electrical Engineering (Archiv fur Elektrotechnik)*, 57(5):281–289, 1975.

[12] U.S. Census Bureau. Fact for Features : 20th Anniversary of Americans with Disabilities Act, 2010.

[13] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm, 2001.

[14] S. Choi and H.Z. Tan. Toward realistic haptic rendering of surface textures. In *ACM SIGGRAPH 2005 Courses*, page 125. ACM, 2005.

[15] C. Colwell, H. Petrie, D. Kornbrot, A. Hardwick, and S. Furner. Haptic virtual reality for blind computer users. In *Proceedings of the third international ACM conference on Assistive technologies*, pages 92–99. ACM, 1998.

[16] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge Univ Pr, 2000.

[17] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1052–1067, 2007.

[18] M. de Pascale, G. de Pascale, D. Prattichizzo, and F. Barbagli. The Haptik Library: a Component based Architecture for Haptic Devices Access. *Proceedings of Eurohaptics 2004*, pages 44–51.

[19] N. Diolaiti and C. Melchiorri. Haptic tele-operation of a mobile robot. In *Robot control 2003 (SYROCO'03): a proceedings volume from the 7th IFAC Symposium, Wrocław, Poland, 1-3 September 2003*, page 7. Pergamon Pr, 2004.

[20] M. Fine and A. Asch. Disability beyond stigma: Social interaction, discrimination, and activism. *Journal of Social Issues*, 44(1):3–21, 1988.

[21] C.P. Gharpure and V.A. Kulyukin. Robot-assisted shopping for the blind: issues in spatial cognition and product selection. *Intelligent Service Robotics*, 1(3):237–251, 2008.

[22] gl.tter. WiiYourself! - native C++ Wiimote library v1.15. http://wiiyourself.gl.tter.org/.

[23] D.H. Grollman and O.C. Jenkins. Dogged learning for robots. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2483–2488. IEEE, 2007.

[24] Stefan Gumhold, Zachi Kami, Martin Isenburg, and Hans-Peter Seidel. Predictive point-cloud compression. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[25] J. Haan and F.C. Park. srLib-SNU Robot Dynamics Library. http://r-station.co.kr/srlib/.

[26] M.T. Hagan, H.B. Demuth, and O.D. Jesús. An introduction to the use of neural networks in control systems. *International Journal of Robust and Nonlinear Control*, 12(11):959–985, 2002.

[27] B. Hannaford and Jee-Hwan Ryu. Time-domain passivity control of haptic interfaces. *Robotics and Automation, IEEE Transactions on*, 18(1):1 –10, feb 2002.

[28] R. Held. Telepresence. *The Journal of the Acoustical Society of America*, 92:2458, 1992.

[29] Marion A. Hersh. *Assistive Technology for Visually Impaired and Blind People*. Springer, 2007.

[30] D. Hong, S. Kimmel, R. Boehling, N. Camoriano, W. Cardwell, G. Jannaman, A. Purcell, D. Ross, and E. Russel. Development of a semi-autonomous vehicle operable by the visually-impaired. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 539–544. IEEE, 2008.

[31] B. Horan, Z. Najdovski, S. Nahavandi, and E. Tunstel. Haptic control methodologies for telerobotic stair traversal. *Int. J. of Intelligent Control and Systems*, 13(1):3–14, 2008.

[32] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *ARTIFICAL INTELLIGENCE*, 17:185–203, 1981.

[33] A. M. Howard and C. H. Park. Haptically guided teleoperation for learning manipulation tasks. In *Proceedings of Robotics: Science and Systems: Workshop on Robot Manipulation*.

[34] Ayanna M. Howard, Chung Hyuk Park, and Sekou Remy. Using haptic and auditory interaction tools to engage students with visual impairments in robot programming activities. *IEEE Transactions on Learning Technologies*, 5:87–95, 2012.

[35] X. Huo, J. Wang, and M. Ghovanloo. Introduction and preliminary evaluation of the Tongue Drive System: wireless tongue-operated assistive technology for people with little or no upper-limb function. *Journal of rehabilitation research and development*, 45(6):921, 2008.

[36] H. Iwata and H. Noma. Volume haptization. In *Virtual Reality, 1993. Proceedings., IEEE 1993 Symposium on Research Frontiers in*, pages 16–23. IEEE, 2002.

[37] H. Jin, S. Soatto, and A.J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63(3):175–189, 2005.

[38] T. Kanade, H. Kano, , and S. Kimura. Development of a video-rate stereo machine. In *Image Understanding Workshop*, Monterey,CA.

[39] J. Karathanasis, D. Kalivas, and J. Vlontzos. Disparity Estimation Using Block Matching and Dynamic Programming. In *Electronics, Circuits, and Systems, 1996. ICECS '96., Proceedings of the Third IEEE International Conference on*, volume 2, pages 728–731.

[40] O. Khatib. Mobile manipulation: The robotic assistant. *Robotics and Autonomous Systems*, 26(2-3):175–183, 1999.

[41] S.Y. Ko, H. Joo, D.G. Choi, H. Kim, and I.S. Kweon. Intelligent Grasping by Learning from Observation. In *Ubiquitous Robots and Ambient Intelligence (URAI 2008), The 5th International Conference on*, 2008.

[42] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran. RFID in robot-assisted indoor navigation for the visually impaired. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 1979–1984. Ieee, 2005.

[43] S. Lee, G. Sukhatme, G. J. Kim, and C. M. Park. Haptic teleoperation of a mobile robot: a user study. *Presence: Teleoperators & Virtual Environments*, 14(3):345–365, 2005.

[44] J. Liu, SC Cramer, and DJ Reinkensmeyer. Learning to perform a new movement with robotic assistance: comparison of haptic guidance and visual demonstration. *Journal of neuroengineering and rehabilitation*, 3(1):20, 2006.

[45] S.A. Ludi and T. Reichlmayr. Developing inclusive outreach activities for students with visual impairments. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, pages 439–443. ACM, 2008.

[46] Stephanie Ludi. ImagineIT: Resources. http://www.se.rit.edu/∼imagine-it/resources.html.

[47] K. Lundin, B. Gudmundsson, and A. Ynnerman. General proxy-based haptics for volume visualization. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 557–560. IEEE, 2005.

[48] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber. A system for robotic heart surgery that learns to tie knots using recurrent neural networks. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 543–548. IEEE, 2007.

[49] R. McGill, J.W. Tukey, and W.A. Larsen. Variations of box plots. *American Statistician*, 32(1):12–16, 1978.

[50] F. Michaud, P. Boissy, H. Corriveau, A. Grant, M. Lauria, D. Labonte, R. Cloutier, M. Roux, M. Royer, and D. Iannuzzi. Telepresence robot for home care assistance. In *Procceedings of AAAI*, 2006.

[51] F. Michaud, P. Boissy, H. Corriveau, A. Grant, M. Lauria, D. Labonte, R. Cloutier, M.A. Roux, MP Royer, and D. Iannuzzi. Telepresence robot for home care assistance. *Procceedings of AAAI*, 20006, 2006.

[52] W.T. Miller, R.S. Sutton, and P.J. Werbos. *Neural networks for control*. MIT press, 1995.

[53] MobileRobots. Advanced Robot Interface for Applications (ARIA) library. http://robots.mobilerobots.com/wiki/ARIA.

[54] M.C. Nechyba and Y. Xu. Human skill transfer: neural networks as learners and teachers. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 314–319. IEEE, 2002.

[55] M.N. Nicolescu and M.J. Mataric. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 241–248. ACM, 2003.

[56] R. Nock and F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1452–1458, 2004.

[57] National Fedration of the Blind. General Braille Information Center. http://nfb.org/braille-general.

[58] A.M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[59] World Health Organization. Fact sheet: Visual Impairment and Blindness. http://www.who.int/mediacentre/factsheets/fs282/en/index.html, 2011.

[60] N. Ouramdane, F. Davesne, S. Otmane, and M. Mallem. 3D interaction technique to enhance telemanipulation tasks using virtual environment. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5201–5207. IEEE, 2007.

[61] C. H. Papadimitriou and K. Steiglitz. 6.1 The Max-Flow, Min-Cut Theorem. pages 120–128, 1998.

[62] M. Pardowitz and R. Dillmann. Towards life-long learning in household robots: The piagetian approach. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 88–93. IEEE, 2007.

[63] Mariotti S.P.M. Pascolini, D. Global estimates of visual impairment: 2010. 2011.

[64] S.D. Pell, R.M. Gillies, and M. Carss. Relationship between use of technology and employment rates for people with physical disabilities in Australia: implications for education and training programmes. *Disability & Rehabilitation*, 19(8):332–338, 1997.

[65] VV Phansalkar and PS Sastry. Analysis of the back-propagation algorithm with momentum. *Neural Networks, IEEE Transactions on*, 5(3):505–506, 2002.

[66] Beryl Plimmer, Andrew Crossan, Stephen A. Brewster, and Rachel Blagojevic. Multimodal collaborative handwriting training for visually-impaired people. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 393–402, New York, NY, USA, 2008. ACM.

[67] V. Potkonjak, D. Kostic, S. Tzafestas, M. Popovic, M. Lazarevic, and G. Djordjevic. Human-like behavior of robot arms: general considerations and the handwriting task–Part II: the robot arm in handwriting. *Robotics and Computer-Integrated Manufacturing*, 17(4):317–327, 2001.

[68] V. Potkonjak, S. Tzafestas, D. Kostic, G. Djordjevic, and M. Rasic. The handwriting problem [man-machine motion analogy in robotics]. *Robotics & Automation Magazine, IEEE*, 10(1):35–46, 2003.

[69] S. Remy, C. H. Park, and A. M. Howard. Improving the performance of ANN training with an unsupervised filtering method. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 2627–2633. IEEE, 2009.

[70] J.E. Robinson. Access to employment for people with disabilities: findings of a consumer-led project. *Disability & rehabilitation*, 22(5):246–253, 2000.

[71] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, page 213, 2002.

[72] Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. Haptic interaction in virtual environments. In *In Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 128–133, 1997.

[73] Jee-Hwan Ryu, Dong-Soo Kwon, and B. Hannaford. Stable teleoperation with time-domain passivity control. *IEEE Transactions on Robotics and Automation*, 20(2):365 – 373, april 2004.

[74] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 2003.

[75] Kenneth Salisbury, Francois Conti, and Federico Barbagli. Haptic rendering: Introductory concepts, 2004.

[76] S. Schaal. Is imitation learning the route to humanoid robots. *Trends in cognitive sciences*, 3(6):233–242, 1999.

[77] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002.

[78] David W Schloerb. A quantitative measure of telepresence. *Presence: Teleoperators and Virtual Environments*, 4(1):64 – 80, 1995.

[79] D.W. Schloerb. A quantitative measure of telepresence. *Presence: Teleoperators and Virtual Environments*, 4(1):64–80, 1995.

[80] SensAble. OpenHaptics toolkit Academic Edition. http://www.sensable.com/products-openhaptics-toolkit.htm.

[81] Anders Soborg. C++ communication library for NXT 2.0. http://www.norgesgade14.dk/bluetoothlibrary.php.

[82] J. Sun, N.N. Zheng, and H.Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 787–800, 2003.

[83] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1068–1080, 2007.

[84] H.Z. Tan, M.A. Srinivasan, B. Eberman, and B. Cheng. Human factors for the design of force-reflecting haptic interfaces. *Dynamic Systems and Control*, 55(1):353–359, 1994.

[85] I. Ulrich and J. Borenstein. The GuideCane-applying mobile robot technologies to assist the visually impaired. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(2):131–136, 2002.

[86] V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, 2000.

[87] D. Wang, Y. Zhang, and C. Yao. Machine-mediated Motor Skill Training Method in Haptic-enabled Chinese Handwriting Simulation System. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5644–5649. IEEE, 2007.

[88] K. Yamazaki, T. Tsubouchi, and M. Tomono. Modeling and motion planning for handling furniture by a mobile manipulator. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1926–1931. IEEE, 2007.

[89] Y. Yokokohji, R.L. Hollis, T. Kanade, K. Henmi, and T. Yoshikawa. Toward machine mediated training of motor skills. Skill transfer from human to human via virtual environment. In *Robot and Human Communication, 1996., 5th IEEE International Workshop on*, pages 32–37. IEEE, 2002.